



Norbert Breier, Peer Stechert, Thomas Wilke (Hrsg.)

Informatik erweitert Horizonte

15. GI-Fachtagung „Informatik und Schule“ – INFOS 2013

26.-28. September 2013 in Kiel

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Proceedings

Series of the Gesellschaft für Informatik (GI)

Volume P-219

ISBN 978-3-88579-613-8

ISSN 1617-5468

Volume Editors

Prof. Dr. Norbert Breier

Universität Hamburg

Fakultät für Erziehungswissenschaft

20146 Hamburg, Germany

E-Mail: norbert.breier@uni-hamburg.de

Dr. Peer Stechert

Regionales Berufsbildungszentrum Technik Kiel

24143 Kiel, Germany

E-Mail: pstechert@rbz-technik.de

Prof. Dr. Thomas Wilke

Christian-Albrechts-Universität zu Kiel

Institut für Informatik

24118 Kiel, Germany

E-Mail: thomas.wilke@email.uni-kiel.de

Series Editorial Board

Heinrich C. Mayr, Alpen-Adria-Universität Klagenfurt, Austria

(Chairman, mayr@ifit.uni-klu.ac.at)

Dieter Fellner, Technische Universität Darmstadt, Germany

Ulrich Flegel, Hochschule für Technik, Stuttgart, Germany

Ulrich Frank, Universität Duisburg-Essen, Germany

Johann-Christoph Freytag, Humboldt-Universität zu Berlin, Germany

Michael Goedicke, Universität Duisburg-Essen, Germany

Ralf Hofestädt, Universität Bielefeld, Germany

Michael Koch, Universität der Bundeswehr München, Germany

Axel Lehmann, Universität der Bundeswehr München, Germany

Peter Sanders, Karlsruher Institut für Technologie (KIT), Germany

Sigrid Schubert, Universität Siegen, Germany

Ingo Timm, Universität Trier, Germany

Karin Vosseberg, Hochschule Bremerhaven, Germany

Maria Wimmer, Universität Koblenz-Landau, Germany

Dissertations

Steffen Hölldobler, Technische Universität Dresden, Germany

Seminars

Reinhard Wilhelm, Universität des Saarlandes, Germany

Thematics

Andreas Oberweis, Karlsruher Institut für Technologie (KIT), Germany

© Gesellschaft für Informatik, Bonn 2013

printed by Köllen Druck+Verlag GmbH, Bonn

Vorwort

Die Fachtagung „Informatik und Schule“ (INFOS) der Gesellschaft für Informatik e. V. (GI) fördert traditionsgemäß den Erfahrungsaustausch zwischen Lehrenden und Forschenden und bietet Anregungen dafür, wie erfolgreicher Informatikunterricht gestaltet werden kann. Sie wird in diesem Jahr zum 15. Mal ausgerichtet. Nach Berlin (1984), Kaiserslautern (1986), München (1989), Oldenburg (1991), Koblenz (1993), Chemnitz (1995), Duisburg (1997), Potsdam (1999), Paderborn (2001), München (2003), Dresden (2005), Siegen (2007), Berlin (2009) und Münster (2011) ist Kiel der Tagungsort. Die Fachtagung steht in diesem Jahr unter dem Motto „Informatik erweitert Horizonte“.

Die INFOS soll eine zeitgemäße qualifizierte informatische Bildung in der Schule auf vielfältige Weise befördern. So besteht das Programm in diesem Jahr aus vier Vorträgen eingeladener Referentinnen und Referenten sowie vierzehn ausgewählten Forschungs-, Entwicklungs- und Erfahrungsbeiträgen, die vom Programmkomitee aus 34 eingereichten Beiträgen ausgewählt wurden. Diese Beiträge beleuchten die folgenden Schwerpunkte der Tagung aus unterschiedlichen Perspektiven: Informatik im Kontext, Schüler- vorstellungen, Informatikunterricht in der Sekundarstufe I, Schülerlabore, Lernchancen im Informatikunterricht, Roboter im Informatikunterricht sowie Studium und Fortbildung. Eine Abrundung erfährt die Konferenz durch insgesamt 18 Workshops, ein Forum für Doktorandinnen und Doktoranden, ein Forum für Lehrkräfte im Ausbildungsstadium, Berichte aus der schulischen Praxis, Poster und eine Ausstellung.

Gemeinsam mit der INFOS wurden in der Vergangenheit häufiger andere Veranstaltungen ausgerichtet, die sich im Umfeld von Schule und Informatik bewegen. Im Jahr 2013 ist das erneut der Fall: Der dritte Tag der INFOS ist gleichzeitig die Fachtagung der GI-Fachgruppe für Informatiklehrerinnen und -lehrer in Schleswig-Holstein und Hamburg einerseits und der erste schleswig-holsteinische Landesfachtag Informatik andererseits.

Als Vorsitzende des Programmkomitees und Ausrichter möchten wir zuvorderst den Mitgliedern unseres Programmkomitees für die sehr schwierige und zeitaufwändige Arbeit, die Einreichungen kritisch durchzusehen und vierzehn für ein interessantes Tagungsprogramm auszuwählen, danken.

Der diesjährigen INFOS wird Glanz durch die neuen Räumlichkeiten des Regionalen Berufsbildungszentrum Technik der Landeshauptstadt Kiel, in denen sie stattfindet, verliehen. Für die Möglichkeit, die INFOS dort auszurichten, und für die Unterstützung, die wir von seiner Seite erfahren haben, danken wir dem Schulleiter, Jürgen Ströh, herzlich.

Ein besonderer Dank gebührt den beiden Organisationen, die für ein schulisches Informatikgroßereignis in Kiel sorgen, indem sie mit uns gemeinsam tagen: der Fachgruppe SH-HILL der GI und dem Institut für Qualitätsentwicklung an Schulen

Schleswig-Holstein. Die Kooperation mit Torsten Otto und Petra Fojut hätte nicht besser sein können.

Zu guter Letzt gilt unser besonderer Dank dem Fachausschuss „Informatische Bildung in Schulen“ und der Geschäftsstelle der GI für die Unterstützung und das in uns gesetzte Vertrauen.

Kiel und Hamburg, im September 2013

Norbert Breier
Peer Stechert
Thomas Wilke

Programmkomitee

Norbert Breier, Hamburg (stv. Vorsitz)

Ira Diethelm, Oldenburg

Steffen Friedrich, Dresden

Lutz Hellmig, Rostock

Henry Herper, Magdeburg

Peter Hubwieser, München

Ludger Humbert, Wuppertal

Detlef Kähler, Kiel

Johannes Magenheim, Paderborn

Torsten Otto, Hamburg

Monika Seiffert, Hamburg

Peer Stechert, Kiel (Vorsitz)

Thomas Wilke, Kiel

Helmut Witten, Berlin

Inhaltsverzeichnis

Vorträge Eingeladener

Beat Döbeli Honegger

Informatik ist mehr als Informatik! Oder: Warum sich die Informatik mit dem Leitmedienwechsel befassen muss..... 11

Ingrid Schirmer

Technik „wofür“ anstelle Technik „an sich“ – Kontextorientierung als Rahmen für genderbewußten Informatikunterricht..... 21

Jan Vahrenhold

Diagnose und individuelle Förderung (im Informatik-Unterricht)..... 23

Tagungsvorträge

Informatik im Kontext

Kerstin Strecker

Medizinische Informatik in der Sek 1 25

Malte Buchholz

Das 3x3-Schema zur Reflexion kontextorientierter Informatik-Unterrichtsgestaltung 35

Schülervorstellungen

Oliver Seifert, Tony Sauck, Maximilian Schwarzbach, Christopher Lerch, Martin Weinert, Maria Knobelsdorf

„Ich glaube, Google ist so was wie eine Vorhalle des Internets“ – Erste Ergebnisse einer qualitativen Untersuchung von Schülervorstellungen von der Suchmaschine Google 45

Magnus Rabel

Grundvorstellungen zu Klassen und Objekten..... 57

Informatikunterricht in der Sekundarstufe I

Sönke Schulmeister

Movingart – ein fächerübergreifendes Kunst-Informatik-Projekt in der Sekundarstufe I..... 67

Katrin Büttner, Gregor Damnik, Steffen Friedrich, Thomas Knapp, Holger Rohland, Martina Rätz

Informatische Kompetenzen im Gebiet „Datenbanken“ – eine Lernstandsanalyse in Sachsen und Thüringen 77

Lernchancen im Informatikunterricht

Jens Gallenbacher, Dominik Heun

Ein moderner Ansatz für Anchored Instruction im Informatikunterricht 87

Hanno Schauer

Debugging-Aufgaben – Programmfehler als Lernchance..... 97

Schülerlabore

Nadine Bergner, Tim Schellartz, Ulrik Schroeder

Informatik und Mathematik – kombiniert im Schülerlabor-Modul „Einstieg in die Computergrafik“..... 107

Jonas Neugebauer, Dieter Engbring

Konzeption eines Informatik-Moduls für ein Schülerlabor 117

Roboter im Informatikunterricht

Lars Otten

Vermittlung von Konzepten der Informationstechnik in Oberstufen der beruflichen Bildung technischer Fachrichtungen unter Verwendung mobiler Roboter als Lernträger127

Mareen Przybylla, Ralf Romeike

Physical Computing im Informatikunterricht137

Studium und Fortbildung

Christian Götz, Peter Hubwieser

Belebt die Einführung des Schulfachs Informatik die Nachfrage nach einem Informatikstudium?147

Claudia Hildebrandt, Ira Diethelm

Eine wirksame Langzeitfortbildung für Informatik-Lehrkräfte157

Verzeichnis der Autorinnen und Autoren167

Informatik ist mehr als Informatik! Oder: Warum sich die Informatik mit dem Leitmedienwechsel befassen muss

Beat Döbeli Honegger
Institut für Medien und Schule (IMS)
Pädagogische Hochschule Schwyz
Zaystrasse 42
CH-6410 Goldau
beat.doebeli@phsz.ch

Abstract: Informatik ist zur Leitwissenschaft der Informationsgesellschaft des 21. Jahrhunderts geworden. Trotzdem widerspiegelt sich diese Bedeutung bisher nicht in der Bildung: Informatik hat es als Thema noch immer sehr schwer in der Schule. Der Beitrag stellt als erstes das Konzept des durch die Informatik ausgelösten Leitmedienwechsels vor. Danach wird mit der Leitmedienwechselreaktionsskala eine Palette an bildungspolitischen Reaktionsmöglichkeiten präsentiert. Daraus wird zum Schluss abgeleitet, dass sich die Informatikdidaktik vermehrt mit ihrem Umfeld beschäftigen muss, um bildungspolitisch Erfolg zu haben.

1 Es braucht ein Fach Informatik...

Seit mindestens 40 Jahren wird in den deutschsprachigen Ländern Deutschland, Österreich und Schweiz versucht, Grundlagen der Wissenschaft Informatik in die Schule zu bringen und zu diesem Zweck in den Lehrplänen zu verankern. In der Schweiz klang dies 1978 etwa so [Mo78]:

Die Informatik [...] ist eine selbständige, schon recht verbreitete Wissenschaft. Ihre Methoden lassen sich auf eine ganze Reihe von Wissensgebieten anwenden, unter denen mehrere unsere Schul-Programme betreffen. Die Verfasser des vorliegenden Berichts betrachten es deshalb als dringlich und wichtig, die Schüler der Oberstufe der Mittelschule in die Informatik einzuführen.

Der Erfolg blieb aber bisher bescheiden. 35 Jahre später klingen entsprechende Publikationen nicht viel anders [KSZ13]:

Das Projekt der Hasler Stiftung «Informatik in der Bildung» zielt auf die Einführung der Informatik als obligatorisches gymnasiales Grundlagenfach. Das vorliegende Buch legt die Argumente für die Notwendigkeit eines solchen Faches umfassend dar.

Mochte vor 40 Jahren noch praktisch niemand glauben, dass automatisierte Informationsverarbeitung eine grössere Bedeutung erlangen und damit zum Gegenstand von Allgemeinbildung werden könnte, so finden sich heute nur noch wenige, welche die

Bedeutung von ICT für alle Lebensbereiche bestreiten würden. Zum Erstaunen einiger Informatiker hat diese immense Bedeutungszunahme aber keineswegs dazu geführt, dass Informatik breiten Eingang in die Lehrpläne oder die Studentafeln gefunden hätte. Selbst die Klage der Informatikdidaktik über diesen Zustand dauert schon mehr als zehn Jahre, schreiben doch z.B. Hartmann und Nievergelt bereits 2002:

„Der ‚Informatikunterricht‘ steckt trotz 40jähriger Geschichte immer noch in den Anfangsgründen und ist noch weit davon entfernt, sich wie etwa Mathematik oder Geschichte als allgemeinbildendes Fach zu etablieren.“ [HN02]

2 Die neue Bildungskrise

Es war ein deutscher Informatikprofessor, der 1982 - also vor mehr als 30 Jahren - auf die notwendigen Veränderungen des Bildungswesens aufgrund der zunehmenden Verbreitung der Informationstechnologie hingewiesen hat. Aus einer technisch-ökonomischen Perspektive machte Klaus Haefner in seinem Buch „Die neue Bildungskrise“ [Ha82] ein paar radikale und weitreichende Aussagen zu den Folgen der zunehmenden Computerisierung:

- **Automatisierung des Automatisierbaren:** Aufgrund ökonomischer Überlegungen ist es unausweichlich, dass Prozesse, deren Automatisierung sich finanziell gesehen lohnt, auch automatisiert werden (a00118).
- **Automatisierung erhöht die Arbeitslosigkeit:** Die Automatisierung wird Arbeitsplätze vernichten und damit die Arbeitslosigkeit erhöhen (a00833).

Laut Haefner lassen sich Berufsbilder angesichts der Computerisierung in drei Gruppen einteilen: **Autonome** (w01448) sind gemäss Haefner Menschen, deren Beruf sich weitgehend ohne Computer ausüben lässt. Als Beispiel nennt Haefner 1982 die Bauern. **Substituierbare** (w01449) sind diejenigen Berufsbilder, die zunehmend durch Computer ersetzt werden. Als **Unberechenbare** (w01450) werden von Haefner schliesslich diejenigen Berufe bezeichnet, die sich nicht durch Computer ersetzen lassen, weil ihre Tätigkeiten nicht berechenbar und damit nicht automatisierbar sind. Haefner nennt hier als Beispiele Ärzte und Lehrpersonen.

Die erste Forderung Haefners nach „angemessener Einführung in die Informationsverarbeitung“ [Ha82] kommt für einen Informatikprofessor nicht überraschend. Die darauf aufbauende Forderung nach einer 1:1-Ausstattung mit persönlicher Informationstechnik ist aber doch erstaunlich für 1982, dem Jahr der Markteinführung des ersten IBM-PCs.

Weit unerwarteter ist jedoch eine andere Schlussfolgerung Haefners:

Das Bildungswesen muss sich intensiv bemühen, Qualifikationen des Menschen zu entwickeln, die deutlich jenseits der Möglichkeiten der Informationstechnik liegen. ([Ha82] S. 267)

Ein Informatikprofessor fordert 1982 aufgrund des Eintritts in die Informationsgesellschaft die Fokussierung auf das Nichtautomatisierbare (a00119)! Im Buch ist die Rede von *Intuition, Kreativität* und *Sinnlichkeit* (S.262-273). Gerade die Automatisierung des

Automatisierbaren verlangt gemäss Haefner im Bildungswesen nach einer Fokussierung auf das Nichtautomatisierbare (a00514). Haefners Buch ist damit ein früher Beleg für die Ansicht, dass die Informationsgesellschaft Auswirkungen auf die Bildung haben sollte, die weit über die Einführung eines Faches Informatik hinaus reichen.

3 Es ist ein Leitmedienwechsel

Zahlreiche Expertinnen und Experten aus verschiedensten Wissenschaftsdisziplinen haben in den letzten 40 Jahren die Erfindung des Computers als massgeblichen Auslöser eines Wandels beschrieben, der mit der Einführung der Schrift oder des Buchdrucks vergleichbar ist und darum vielerorts als *Leitmedienwechsel* bezeichnet wird:

- Nicolas Negroponte beschreibt in seinem Buch „*total digital*“ die weit reichenden Konsequenzen des Wandels „*vom Atom zum bit*“ [Ne95].
- Der Medientheoretiker Michael Giesecke beschäftigt sich in seinem Werk „*Von den Mythen der Buchkultur zu den Visionen der Informationsgesellschaft*“ [Gi02] mit den Auswirkungen des Buchdrucks auf Wirtschaft, Gesellschaft und Bildung. Gemäss Giesecke kann ein Wechsel des Leitmediums vom Buch zum Computer nicht erfolgen, ohne dass viele bestehende Organisationsstrukturen und Abläufe in Frage gestellt und umgestaltet werden.
- Der Ökonom Thomas Friedman weist in seinem Buch „*Die Welt flach*“ [Fr05] darauf hin, dass Computer und Internet eine dritte Stufe der Globalisierung ermöglicht hätten: Mit der Entdeckung Amerikas sei die Globalisierung 1.0 durch Staaten erfolgt. Vom 18. bis zum 20. Jahrhundert folgte die Globalisierung 2.0 durch Unternehmen. Seit dem Jahr 2000 ist nun die Globalisierung 3.0 durch einzelne Menschen im Gang.
- Der Soziologe Dirk Baecker vertritt in seinem Buch „*Studien zur nächsten Gesellschaft*“ [Ba07] die These, dass jede neue Kommunikationsmöglichkeit des Menschen zu neuen Gesellschaftsformen geführt habe, da die bisherige Gesellschaftsform mit den überschüssigen Kommunikationsmöglichkeiten überfordert gewesen sei. So sei die Sprache der Auslöser für die Stammesgesellschaft, die Schrift der Auslöser für antike Hochkulturen und der Buchdruck der Auslöser der modernen Gesellschaft gewesen. Dementsprechend werde der Computer zum Auslöser einer neuen Gesellschaftsform werden.
- In ihrem aktuellen Buch „*Race against the machine*“ [BM11] weisen die beiden Ökonomen Erik Brynjolfsson und Andrew McAfee 2011 auf die bereits von Haefner beschriebene Gefahr der Arbeitsplatzvernichtung durch Computer hin.
- Im deutschsprachigen Raum hat in den letzten Jahren unter anderem der ehemalige IBM-Cheftechnologe Gunter Dueck auf die tiefgreifenden Veränderungen aufgrund der Digitalisierung hingewiesen (z.B. im Re:publica-Vortrag von 2011 [Du11]).

Selbstverständlich ist die hier eher holzschnittartig präsentierte These, die digitale Informationsverarbeitung sei der Auslöser eines Leitmedienwechsels nicht unumstritten. So weist etwa Steinbicker darauf hin, dass der Begriff und wesentliche Aspekte der Idee einer Informationsgesellschaft bereits in den 1960er Jahren formuliert worden seien, also vor

der massiven Zunahme der digitalen Medien [St01]. Seidensticker wiederum zweifelt im Buch „FutureHype“ daran, dass es sich tatsächlich um eine Revolution handle und versucht dies mit neun Thesen nachzuweisen [Se06]. Für die nachfolgenden Überlegungen möchte ich jedoch von einem Leitmedienwechsel ausgehen und als nächstes genauer betrachten, wie die Bildung darauf reagieren soll (f00154).

4 Wie soll die Schule auf den Leitmedienwechsel reagieren?

Während die meisten eine gesellschaftliche und wirtschaftliche Veränderung aufgrund digitaler Technologien als gegeben akzeptieren, gehen die Meinungen weit auseinander, wie die Bildung, insbesondere die Schule auf den Leitmedienwechsel reagieren soll. Es lassen sich mindestens acht unterschiedliche Reaktionsstufen unterscheiden, geordnet nach zunehmender Veränderung gegenüber der bisherigen Organisation von Schule:

a) Reaktionsstufe 0: Gar nicht reagieren. (a01180)

Nach dem Motto: „*Die Schülerinnen und Schüler sollen erst richtig lesen, schreiben und rechnen lernen!*“ wird der Leitmedienwechsel bei der Planung zukünftiger Schule schlicht ignoriert. In der Schweiz ist in der Diskussion um den kommenden, sprachregionalen Lehrplan 21 der (chancenlose) Alternativvorschlag der Schweizerischen Volkspartei ein Beispiel für diese Reaktion [SVP10].

b) Reaktionsstufe 1: Integration in alle Fächer (a01181)

Die Begründungen zur Integration in alle Fächer reichen von einem eher sanften: „**Es reicht, wenn sich alle Fächer etwas mit digitalen Themen beschäftigen**“ bis zum stärkeren: „*Da der Leitmedienwechsel überall stattfindet, müssen entsprechende Themen auch überall integriert werden.*“. Zahlreiche Schweizer Lehrpläne verfolgen diese Strategie.

c) Reaktionsstufe 2: Es braucht ein eigenes Fach (a01182)

Kritiker der Integrationsvariante monieren, dass damit die Verbindlichkeit fehle. Erst ein eigenes Fach mit Zeitfäch und Noten sorge für die notwendige Verbindlichkeit.

d) Reaktionsstufe 3: Es braucht beides (a01183)

Während bei der Integration in alle Fächer die Verbindlichkeit fehlt, droht bei der Errichtung eines eigenen Fachs das Thema an eine Speziallehrperson ausgelagert zu werden. Die anderen Lehrpersonen haben damit eine Ausrede, sich nicht mehr damit befassen zu müssen. Aus diesem Grund ist sowohl ein Fach als auch die Integration in alle anderen Fächer notwendig. Es zweifelt niemand an einem Schulfach Deutsch, auch wenn in allen anderen Schulfächern ebenfalls Deutsch gesprochen wird.

e) Reaktionsstufe 4: Wer redet noch von Fächern? (a01184)

Der Leitmedienwechsel ist geprägt von Konvergenz, die Welt ist vernetzt und globalisiert. Damit sind Fächergrenzen interdisziplinären Problemlösungen im Weg und müssen beseitigt werden. Die Schule der Zukunft benötigt nicht mehr von bisherigen Lösungen, sondern neue Lösungen in Form von vernetztem Lernen.

f) Reaktionsstufe 5: Wer redet noch von Schule? (a01185)

Die heutige Idee der Schule ist geprägt von der Industriegesellschaft (Lernfabrik). Wir müssen uns von der bisherigen Idee des rein formalen Lernens verabschieden:

„Computer und Internet ermöglichen ein individuelles Lernen ohne Lernfabriken.“
 Prominenter und früher Vertreter dieser Haltung war Seymour Papert [Pa84].

g) **Reaktionsstufe 6: Wer redet noch von Bildung? (a01186)**

Die radikalste Reaktionsstufe schliesslich orientiert sich an Ray Kurweils These des *singularity points*, d.h. dem Zeitpunkt an welchem die Rechenleistung aller Computer auf der Welt die Rechenleistung eines menschlichen Gehirns erreicht. Spätestens zu diesem Zeitpunkt werde es nicht mehr darum gehen, die Bildung von Menschen zu optimieren, sondern das Zusammenleben von Menschen mit intelligenten Robotern.

Neben diesen Reaktionsstufen gibt es noch eine weitere, schwieriger einzuordnende Reaktionsweise:

h) **Reaktionsstufe -1: Gegensteuern! (a01192)**

Die Schule muss ein Gegengewicht zum Leitmedienwechsel bieten und diejenigen Aspekte der Welt betonen, die durch den Leitmedienwechsel bedroht sind und/oder wichtiger werden, weil sie sich nicht digitalisieren/automatisieren lassen.

Schwieriger einzuordnen, weil sie einerseits von Experten wie Haefner oder Dueck vertreten wird, welche dem Leitmedienwechsel durchaus positiv gegenüberstehen, andererseits aber auch von Personen und Gruppierungen, die das Rad der Zeit gerne anhalten bzw. zurückdrehen würden. Gerade die „Reaktionsstufe -1“ zeigt, dass diese Reaktionsweisen sich nicht ganz so linear anordnen lassen, wie dies Abbildung 1 suggeriert.

Die Leitmedienwechselreaktionsskala illustriert das breite Spektrum an möglichen Reaktionen auf die durch Informationstechnik ausgelösten Veränderungen in allen Bereichen. Die Informatikdidaktik muss sich des bildungspolitischen Umfelds bewusst sein, in welchem sie ihre Forderungen stellt. Sie muss somit die Leitmedienwechselreaktionsskala und den damit verbundenen Abstraktionsgrad der bildungspolitischen Diskussion berücksichtigen, wenn sie ihre Anliegen wirkungsvoll einbringen will. Der Leitmedienwechsel stellt Fragen weit grösserer Tragweite als diejenige nach einem Fach Informatik. Werden diese Fragen von der Informatikdidaktik negiert oder ignoriert, so sinkt damit die bildungspolitische Akzeptanz ihrer Anliegen.



Abbildung 1: Die Leitmedienwechselreaktionsskala

5 Sichtweisen der Informatik

5.1 Wie definieren die Informatiker die Informatik

Eine weitere mindestens 40-jährige Tradition der Informatik(-didaktik) besteht in der immerwährenden Schwierigkeit, Informatik als Themengebiet zu definieren, eine notwendige Voraussetzung sowohl im bildungspolitischen Diskurs als auch bei der Erstellung von Lehrplänen und Lehrmitteln. Es ist hier nicht der Ort, um vertieft auf die Frage „*Was ist Informatik?*“ (f00140) einzugehen. Es genügt aufzuzeigen, dass sich die Informatik professionsintern seit 40 Jahren kontrovers mit dieser Frage auseinandersetzt (z.B. [Co92], [De89], [De13], [Re10], [Sch07], [Sch13], [We97]), als auch gegen aussen immer wieder versucht, das Themengebiet zu definieren (z.B. [BCM06], [SG08], [KSZ13]) und die beiden informatikdidaktisch prägenden Modelle der *great principles of computing* von Denning [De03] und im deutschsprachigen Raum die *fundamentalen Ideen der Informatik* (w01098) von Schwill zu erwähnen [Sch93].

5.2 Wie denken Informatiker, dass Nichtinformatiker die Informatik sehen

Wenn Informatiker selbst seit mindestens 40 Jahren immer wieder versuchen, ihre eigene Profession zu definieren, so ist es nicht weiter erstaunlich, dass die Öffentlichkeit kein klares Bild der Informatik besitzt. Informatik-Didaktiker weisen seit Jahren auf die Problematik hin, dass die öffentliche Wahrnehmung der Informatik nicht mit dem Selbstbild der Informatik übereinstimmt und interessieren sich auch empirisch für das Fremdbild der Informatik (z.B. [KS07], [Ja13]).

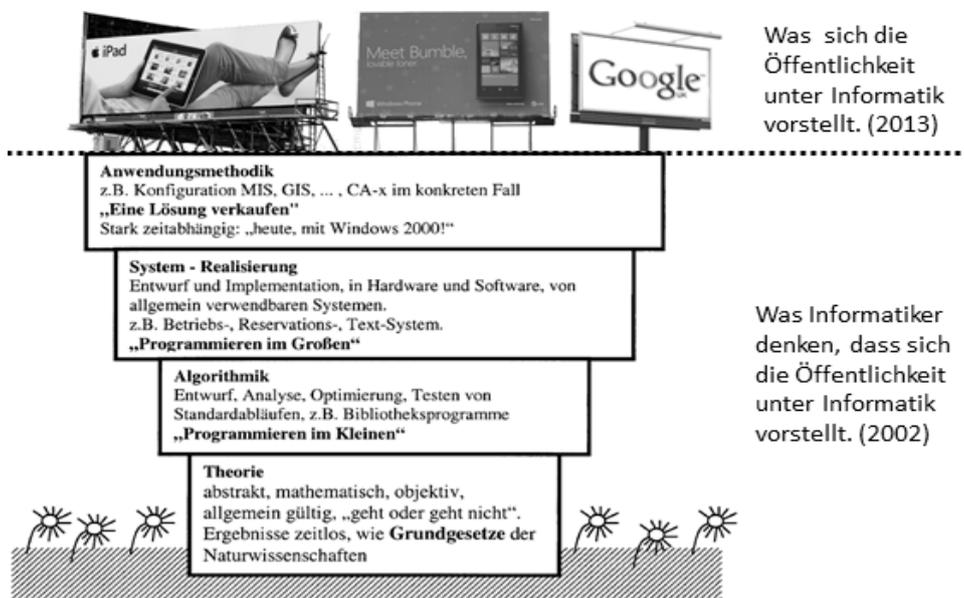


Abbildung 2: Was Informatiker denken, dass sich die Öffentlichkeit unter Informatik vorstellt [HN02] und was sich die Öffentlichkeit vermutlich tatsächlich vorstellt.

Nievergelt prägte 1995 das Bild des Informatikturms ([Ni95] und [HN02]), dessen untere Stockwerke die Öffentlichkeit kaum zur Kenntnis nehmen würde (siehe unteren Teil von Abbildung 2). Bei Nievergelt's Informatikturm sind kommerzielle, in der Arbeitswelt verbreitete Programme an der sichtbaren Oberfläche, die private Nutzung ist in den Publikationen von 1995 und 2002 noch praktisch kein Thema. Aus heutiger Sicht muss auch Nievergelt's Informatikturm zu den Elfenbeintürmen gezählt werden, hat doch in den letzten Jahren die private und ubiquitäre Verbreitung von ICT dazu geführt, dass vermutlich auch das oberste Stockwerk von Nievergelt unter den Wahrnehmungshorizont der Öffentlichkeit verschwunden ist (siehe Abbildung 2). Es ist somit in den letzten 10 Jahren um eine weitere Schicht schwieriger geworden, der Öffentlichkeit ein adäquates Bild der Informatik zu vermitteln.

5.3 Wie Informatik im Schulalltag gesehen (und vermittelt) werden könnte

Informatik(-didaktiker) versuchen oft der unklaren bis verzerrten öffentlichen Wahrnehmung ihrer Profession zu begegnen, indem sie die Bedeutung der „echten“ und „engeren“ Informatik betonen, sich für das oberste Stockwerk des neuen Informatikturms und die damit verbundenen allgemeinbildenden Kompetenzen nicht zuständig erklären und diese im Extremfall sogar abwerten. Dies ist aber weder bildungspolitisch noch didaktisch zielführend. Um Neues zu vermitteln – und das ist sowohl in der Bildungspolitik wie auch in der Informatikdidaktik das Ziel – sollte auf dem Vorwissen und der Lebenserfahrung des Zielpublikums aufgebaut und die eigenen Inhalte im Kontext vermittelt werden. Wie sieht denn heute der Kontext der Informatikdidaktik aus?

Um nicht mehrfach belegte und unterschiedlich definierte Schlagworte verwenden zu müssen, lassen sich die drei für Berufsfähigkeit und mündige Lebensbewältigung notwendigen Kompetenzbereiche bezüglich digitaler Medien folgendermassen umreissen [ICT13]:

- a) Schülerinnen und Schüler verstehen Grundkonzepte der Wissenschaft Informatik und nutzen sie zur Entwicklung von Lösungsstrategien in allen Lebensbereichen
- b) Schülerinnen und Schüler nutzen Informations- und Kommunikationstechnologien in allen Bereichen des Lebens effektiv und effizient
- c) Schülerinnen und Schüler produzieren digitale Inhalte und reflektieren die Nutzung, Bedeutung und Wirkung von (digitalen) Medien kritisch

In der aktuellen schweizerischen bildungspolitischen Diskussion werden diese drei Kompetenzbereiche als

- a) Informatik
- b) Anwendungskompetenzen oder ICT
- c) Medienbildung

bezeichnet. Sowohl im Alltag als auch in der Schule bedingen sich diese Bereiche gegenseitig:

- Ohne Informatik keine Grundlage für das Verständnis von ICT und Medienbildung.
- Ohne Medienbildung fehlen den Anwendungskompetenzen und der Informatik die Reflexion und gesellschaftliche Einbettung.

- Ohne Anwendungskompetenzen und Lebensweltbezug artet sowohl Informatik als auch Medienbildung in Trockenschwimmen aus.

Der genannte Lebensweltbezug besteht derzeit einerseits in der bald flächendeckenden privaten ICT-Infrastruktur und der darauf aufbauenden privaten Nutzung und andererseits die schulische ICT-Infrastruktur und die -Nutzung. Sowohl bildungspolitische als auch fachdidaktische Überlegungen müssen alle drei Themenbereiche und vier Kontextfelder im Blick haben, um die gewünschte Wirkung zu erzielen (siehe Abbildung 3).

Mit dieser Forderung nach einer gesamtheitlichen Betrachtungsweise sollen keinesfalls alle Themenbereiche vermischt und in einen Topf geworfen werden. Informatik soll und muss auf Unterschiede zu Anwendungskompetenzen und Medienbildung hinweisen. Sowohl in Lehrplandiskussionen als auch im Unterrichtsalltag sind diese Aspekte bewusst zu unterscheiden und trotzdem zu verbinden.

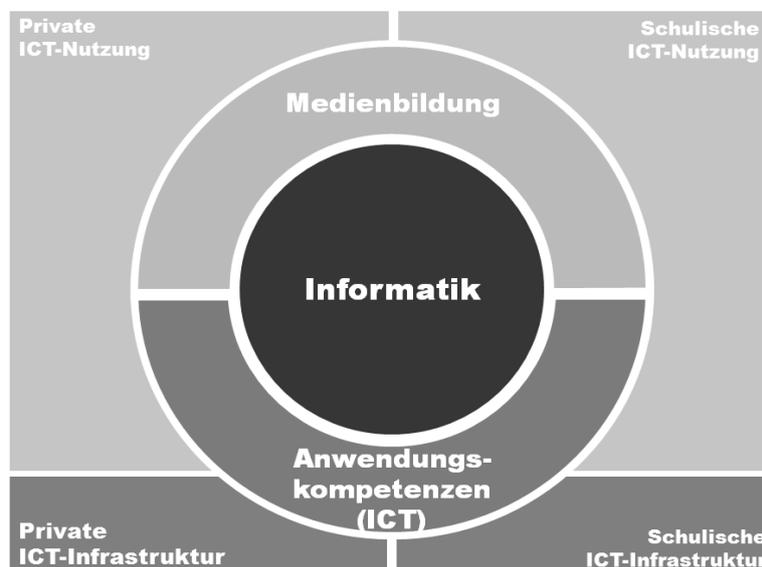


Abbildung 3: Digitale Kompetenzbereiche und Kontext

Informatikdidaktiker dürfen sich nicht auf die Informatik beschränken und sich vor Medienbildung und Anwendungskompetenzen verschliessen. Sie müssen sowohl entsprechende Fragen aus der Bildungspolitik als auch von Schülerseite beantworten und informatische Kompetenzen mit Anwendungskompetenzen und Medienbildung an praktischen Alltagsbeispielen verknüpfen können. Mit weicheren Aspekten als reiner Informatik umzugehen ist für Informatiker zum Teil ähnlich schwierig, wie den Titel dieses Beitrags zu lesen, ohne sich an der Rekursion zu verschlucken. Aber Informatik muss mehr sein als Informatik, wenn sich Informatik in der Bildung durchsetzen soll!

Und weil sowohl Bildungspolitik als auch Unterrichtsalltag oft iterativ ist, hier als Wiederholung meine Empfehlungen von 2010 zur gleichen Problematik [Dö10]:

1. Einige Dich, was Informatik ist
2. Zeige die Bedeutung der Informatik, bleibe aber bescheiden
3. Sei zeitlos in der Begründung und in den Konzepten, aber zeitnah in den Werkzeugen und Beispielen
4. Zeige informatische Konzepte in anderen Fachbereichen
5. Mache Informatik greifbar
6. Vermeide es, Überzeugte überzeugen zu wollen
7. Übe Dich in der Kunst der Iteration

Biblionetzverweise

Alle nach dem Schema (x12345) aufgebauten Verweise im Text verweisen auf entsprechende Webseiten nach dem Schema <http://doebe.li/x12345>

Literaturverzeichnis (auch unter <http://doebe.li/t16000> abrufbar)

- [Ba07] Baecker, D.: Studien zur nächsten Gesellschaft, Suhrkamp Verlag.
<http://doebe.li/b04152>
- [BCM06] Biundo, S.; Claus, V. & Mayr H. C.: Was ist Informatik? GI-Positionspapier.
<http://doebe.li/t07972>
- [BM11] Brynjolfsson, E.; McAfee, A.: Race Against The Machine. How the Digital Revolution is Accelerating Innovation, Driving Productivity, and Irreversibly Transforming Employment and the Economy
<http://doebe.li/b04724>
- [Co92] Coy, W.: Sichtweisen der Informatik. Vieweg Verlag, 1992.
<http://doebe.li/b00148>
- [De89] Denning, P. et al.: Computing as a Discipline. In Communications of the ACM 1/1989.
<http://doebe.li/t05075>
- [De03] Denning, P.: Great Principles of Computing. In: Communications of the ACM 11/2003.
<http://doebe.li/t05077>
- [De13] The Science in Computer Science. In: Communications of the ACM, Vol. 56 No. 5.
<http://doebe.li/t15395>
- [Dö10] Döbeli Honegger, B.: ICT im Hosensack - Informatik im Kopf? Gedanken über ICT und Informatik in der Schweizer Primarschule. In: Brandhofer, G. et al. 25 Jahre Schulinformatik, OCG Verlag.
<http://doebe.li/t12345>
- [DKZ13] Döbeli Honegger, B.; Kuhnt, B. & Zehnder, C. A.: Informatik, ICT und Medienbildung. In: [KSZ13].
<http://doebe.li/t14683>
- [Du11] Dueck, G.: Das Internet als Gesellschaftsbetriebssystem. Vortrag gehalten auf der Re:publica 2011, Berlin.
<http://doebe.li/t12621>
- [Fr05] Friedman, T.: The World is flat. Penguin Books.
<http://doebe.li/b02512>
- [Gi02] Giesecke, M.: „Von den Mythen der Buchkultur zu den Visionen der Informationsgesellschaft“, Suhrkamp Verlag, 2002.
<http://doebe.li/b02961>

- [Ha82] Haefner, K.: Die neue Bildungskrise, Birkhäuser Verlag Basel, 1982.
<http://doebe.li/b00127>
- [HN02] Hartmann, W.; Nievergelt J.: Informatik und Bildung zwischen Wandel und Beständigkeit. Informatik-Spektrum 6/02.
<http://doebe.li/t02299>
- [ICT13] ICTswitzerland: Digitale Kompetenzen benötigen Verbindlichkeit im Lehrplan 21!
<http://doebe.li/t15700> (im Druck)
- [Ja13] Jaglo, M.: Hardwarefreaks und Kellerkinder. Klischeevorstellungen über Informatik und die Auseinandersetzung der Studierenden damit. In: [Sch13]
<http://doebe.li/t15431>
- [KS07] Knobelsdorf, M. & Schulte, C.: Das informatische Weltbild von Studierenden. In: Schubert, S.: INFOS 2007 – Didaktik der Informatik in Theorie und Praxis
<http://doebe.li/t07485>
- [KSZ13] Kohlas, J.; Schmid, J. & Zehnder, C. A.: informatik@gymnasium. NZZ Verlag, 2013
<http://doebe.li/b05092>
- [Mo78] Morel, R. et al.: Die Einführung der Informatik an den Mittelschulen. Informationsbulletin Nr. 13 der EDK
<http://doebe.li/b03462>
- [Ne95] Negroponte, N.: Total digital. Die Welt zwischen 0 und 1 oder Die Zukunft der Kommunikation. Bertelsmann Verlag, 1997 (vergriffen).
<http://doebe.li/b00099>
- [Ni95] Nievergelt, J.: Welchen Wert haben theoretische Grundlagen für die Berufspraxis? Gedanken zum Fundament des Informatik-Turms. Informatik-Spektrum 18(6):342-344
<http://doebe.li/t02492>
- [Pa84] Papert, S.: Trying to predict the future. Popular Computing. Volume, 3. Issue, 13, 1984
<http://doebe.li/t15231>
- [Re10] Rechenberg, P.: Was ist Informatik? In: Informatik Spektrum 1/2010.
<http://doebe.li/t10706>
- [Sch93] Schwill, A.: Fundamentale Ideen der Informatik. In: Zentralblatt für Didaktik der Mathematik, 1993.
<http://doebe.li/t01452>
- [Sch07] Schelhowe, H.: Technologie, Imagination und Lernen. Waxmann Verlag, 2007.
<http://doebe.li/tb03147>
- [Sch13] Schinzel, B.: Weltbilder in der Informatik: Sichtweisen auf Profession, Studium, Genderaspekte und Verantwortung. Spektrum der Informatik, 36(3), Juni 2013
<http://doebe.li/b05219>
- [Se06] Seidensticker, B.: FutureHype - The Myths of Technology Change, Berret-Köhler Publishers, 2006.
<http://doebe.li/b02877>
- [SG08] Schleier, J. & Golliez, A.: Was ist Informatik? Broschüre anlässlich des Schweizer Jahres der Informatik.
<http://doebe.li/t09275>
- [St01] Steinbicker, J.: Zur Theorie der Informationsgesellschaft, leske + budrich Verlag, 2001.
<http://doebe.li/b05273>
- [SVP10] Schweizerische Volkspartei: Der SVP Lehrplan.
<http://doebe.li/b04197>
- [We97] Wegner, P.: Why Interaction is More Powerful Than Algorithms. In: Communications of the ACM, May 1997, Vol.40, No. 5, S. 81 – 91
<http://doebe.li/t02515>

Technik „wofür“ anstelle Technik „an sich“

Kontextorientierung als Rahmen für genderbewußten Informatikunterricht

Ingrid Schirmer

Fachbereich Informatik, ITG
Universität Hamburg
Vogt-Kölln-Str. 30
22527 Hamburg
schirmer@informatik.uni-hamburg.de

Informations- und Kommunikationstechnologien, digitale Wirtschaft und Gesellschaft sind Realität und Teil der Lebenswirklichkeit unserer Schüler. Informatische Bildung – als Teil der Allgemeinbildung und zur Ausbildung von Informatikfachkräften – ist daher wichtiger denn je. Sie bildet die Grundlage zur Gestaltung nicht nur der Technologien, sondern auch der Gesellschaft von heute und morgen. Bedarfe an informatisch Gebildeten sind somit hoch und dringlich, um die Ausgestaltung und Funktionsfähigkeit der digitalen Dimension angesichts steigender Abhängigkeit aller Bereiche des menschlichen Handelns von dieser Dimension in den kommenden Generationen weiterhin zu gewährleisten.

Dennoch haben Schülerinnen und Studentinnen den Weg in die Informatik (nahezu konstant über die vergangenen 40 Jahre) zahlenmäßig nur in geringem Ausmaß gefunden. Hierzu tragen das vorherrschende Frauenbild einerseits und das damit kaum vereinbare Bild der Informatik andererseits bei, die in Medien, Werbung und gesellschaftlichem Normen und Werten in unserem Land tradiert werden. Auch fallen Weichenstellungen in der fachlichen Ausrichtung und Berufswahl gerade in das Alter der Pubertät, das – durch die wachsende Identifikation mit dem eigenen Geschlecht – eine starke Anpassung an gesetzte Verhaltensformen und Klischees und die Orientierung an Gruppierungen des gleichen Geschlechts begünstigt.

Was kann Informatikunterricht – sofern Schülerinnen an diesem überhaupt teilnehmen – gegenüber diesen starken Einflüssen „von außen“ bewirken? Welche inhaltlichen und didaktischen Anreize können Informatikunterricht für Schülerinnen und auch Schülern, die nicht in die Informatik finden, dennoch attraktiv machen?

Ein Schlüssel liegt unseres Erachtens auf dem Fokus, mit dem die Beschäftigung mit Inhalten der Informatik vermittelt wird. Erfahrungswerte zeigen, dass ein Großteil der Schülerinnen und Studentinnen nicht an Technik „an sich“ interessiert sind, sondern an Technik „wofür“. Die Anwendungskontexte und die Gestaltung von Technik auf diese hin motivieren sie, sich mit Technik und den Kontexten gleichermaßen zu befassen.

Menschen zu unterstützen kann auch hier im Zentrum stehen. Kreativität ist erwünscht, Hinwendung, Teamarbeit, all dies ist (er-)lebbar.

Mit „Bringing contexts into the classroom¹“ wird ein kontextorientierter Ansatz vorgestellt, der „reale“ Anwendungskontexte der Alltags- und Berufswelt voranalysiert und Prototypen für den Informatikunterricht zur Verfügung stellt. Auf Basis dieser Analysen und darauf aufbauender Artefakte können sowohl „klassische“ informatische Fähigkeiten erlernt werden – z.B. Programmierung durch Erweiterung der Funktionalitäten des Prototyps – als auch ein Ko-Design der Informatiksysteme und des Kontextes in Teams diskutiert, überlegt, vorgenommen und evaluiert werden.

Eine Diskussion, wie aufwändig derartige Ansätze für den Informatikunterricht sind und wie entsprechende Materialien und Unterrichtseinheiten gesammelt, gewartet und zur Verfügung gestellt (z.B. über MOOC-basierte Fortbildungskurse) werden können, beschließt den Vortrag.

¹ Rick, D; Morisse, M.; Schirmer, I. (2012) **Bringing contexts into the classroom: a design-based approach.** In: Proceedings of the 7th Workshop in Primary and Secondary Computing Education (WiPSCE '12). ACM, New York, S. 105–115. DOI 10.1145/2481449.2481476

Diagnose und individuelle Förderung (im Informatik-Unterricht)

Jan Vahrenhold
Institut für Informatik
Westfälische Wilhelms-Universität Münster
Einsteinstr. 62
48149 Münster
jan.vahrenhold@uni-muenster.de

Abstract: Die Leitideen der Diagnose und individuellen Förderung sind in den vergangenen Jahren nicht nur in den Blickpunkt der bildungswissenschaftlichen und fachdidaktischen Forschung gerückt, sondern haben bereits Eingang in einige lehrerbildende Studiengänge genommen. Ein Beispiel hierfür stellt das Projekt *dortMINT* dar, das im Rahmen der Exzellenzinitiative „MINT-Lehrerbildung“ von der *Deutsche Telekom Stiftung* an der Technischen Universität Dortmund in den Jahren 2010–2012 gefördert wurde. Dieses interdisziplinäre Projekt verfolgte das Ziel, die Voraussetzungen dafür zu schaffen, angehende Lehrerinnen und Lehrer in die Lage zu versetzen, fachlich fundiert eine unterrichtliche Diagnostik so zu betreiben, dass individualisierte Fördermaßnahmen abgeleitet und umgesetzt werden können.

In meinem Vortrag werde ich ausgehend von unseren im Rahmen dieses Projekts gewonnenen Erkenntnissen zunächst einen Überblick über die Leitideen der Diagnose und individuellen Förderung geben. Ich werde mich dann anhand konkreter Fallbeispiele mit den folgenden Überlegungen aus Sicht der Informatik beschäftigen:

- Was können wir von den Erfahrungen anderer Fächer lernen, welche Fehler können wir vermeiden?
- Was können wir von den Erfahrungen der Hochschuldidaktik lernen, welche Fehler können wir vermeiden?
- Was können wir aus der Geschichte der Schulinformatik lernen, welche Fehler können wir vermeiden?

Danksagung Die Arbeiten, die zu den vorgestellten Erkenntnissen führten, wurden u.a. mit Holger Danielsiek, Rebecca Doherty, Ludger Humbert, Maria Knobelsdorf, Arno Pasternak, Wolfgang Paul und Renate Thies durchgeführt und zu Teilen an der Technischen Universität Dortmund durch die *Deutsche Telekom Stiftung* (Projekt *dortMINT*) gefördert.

Medizinische Informatik in der Sek 1

Kerstin Strecker

Hainberg-Gymnasium Göttingen
Friedländer Weg 19
37085 Göttingen
kerstin.strecker@gmx.de

Abstract: In diesem Artikel werden Unterrichtssequenzen aus verschiedenen Inhaltsbereichen der Informatik der Sekundarstufe 1 anhand medizinischer Kontexte vorgestellt. Sie zeigen, wie Informatik unter dem Oberbegriff „Informatik hilft!“ und damit einem sozialen Kontext unterrichtet werden kann. Der soziale Kontext wirft einen anderen Blick auf die Informatik, die oft eher anhand technischer und mathematischer Problemstellungen unterrichtet wird. Dies soll helfen, andere Zielgruppen zu erreichen, insbesondere Mädchen für die Wahl des Faches Informatik zu motivieren.

1 Einführung

An Universitäten werden seit langem immer mehr sogenannte Bindestrich-Informatiken gelehrt wie Bioinformatik, Wirtschaftsinformatik, Geoinformatik oder andere, da Informatik Werkzeuge und Verfahren liefert, Lösungsideen für Probleme der Anwendungsbereiche zu entwickeln und dies nicht ohne spezifische Kenntnisse des Anwendungsbereichs erfolgreich sein kann. Überträgt man dies in die Schule, reden wir von fächerübergreifendem Unterricht oder einer Art von „Informatik im Kontext“ (vgl. [DKW11]), wobei der Kontext nicht nur für einzelne Unterrichtssequenzen besteht, sondern die Unterrichtskontexte ihrerseits zusätzlich fachlich zusammenhängend sind und über den gesamten Zeitraum des Informatikunterrichts hinweg, z.B. in der gesamten Sekundarstufe 1, integrativ unterrichtet werden. „Der Anwendungsbereich [...] liefert automatisch Problemstellungen, Beispiele und sinnvolle Anwendungen für den informatischen Bereich.“ [Mo10]. In diesem Artikel sollen Sequenzen aus unterrichtsrelevanten Bereichen der Informatik der Sek 1 anhand medizinischer Kontexte aufgezeigt werden. Ein Wahlpflichtkurs „Medizinische Informatik“ wurde in den Schuljahren 2007/2008 bis einschließlich 2009/2010 in Jahrgang 7 bis 9 vierstündig pro Woche durchgeführt. Mit dem Anwendungsbereich Medizin wurde bewusst ein sozialer Kontext gewählt, um vor allem Mädchen für eine Wahl von Informatik zu motivieren. Schuster et. al. schreiben: „Die Ausbildungsinhalte und -ziele [...] der Ingenieurwissenschaften sprechen interdisziplinär, sozial und vielseitig interessierte Jugendliche nicht an. Traditionelle ingenieurwissenschaftliche (Hochschul-) Ausbildungen weisen zu wenig Bezüge zu anderen Disziplinen auf und erweisen sich daher als unattraktiv für Frauen.“[SW06]. Sie schreiben weiter: „Mädchen haben also kein prinzipielles Desinteresse an Naturwissenschaften und Technik. Vielmehr verlieren sie ihr Interesse im Laufe der Zeit, da [...] die in der Schule vermittelten Themen ihrem

Lebens- und Relevanzbereich nicht entsprechen.“ [...] „Bisher sind Stoffvermittlung und Inhalte des naturwissenschaftlichen Unterrichtes [...] eher an der Lebenswelt der Jungen orientiert [...]“ [SW06]. Mit dem sozialen Kontext Medizin wird versucht, einen solchen Relevanzbereich zu schaffen. Die geschlechtsspezifische Zusammensetzung des genannten Kurses im Wahlpflichtbereich im Verhältnis zu späteren Informatikkursen ohne Anwendungsfach zeigt Abbildung 1. Obwohl die geringen Zahlen: 28 (2007/08-2009/10) bzw. 25 (2009/10-2011/12) Schülerinnen und Schüler (SuS) keine generellen Aussagen zulassen, zeigt sich trotzdem die Tendenz, dass Mädchen durch das Anwendungsfach Medizin eher zur Wahl eines Informatikkurses motiviert werden konnten.

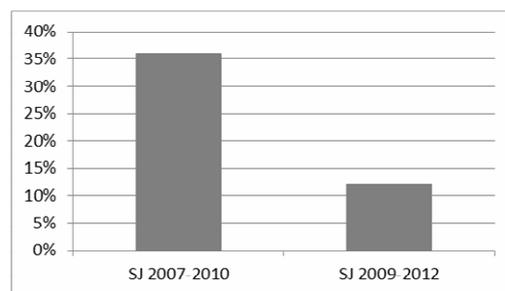


Abbildung 1: Anteil Mädchen im Informatikkurs Medizin-Informatik (SJ 2007/08-2009/10) und im Kurs Informatik ohne Anwendungsfach (SJ 2009/10-2011/12)

2 Die Stellung des Anwendungsbereichs Medizin

Informatik und Medizin können auf verschiedene Weisen miteinander kombiniert werden. Wir teilen die Möglichkeiten grob in drei Kategorien ein, die wir im Folgenden näher erläutern wollen und die alle drei ihren Platz im Unterricht finden sollten.

In der ersten Kategorie steht das Erlernen informatischer Inhalte im Vordergrund, die Medizin liefert lediglich die Umkleidung der Aufgabe. Beispiele: Die Modellierung der Funktionalität technischer Systeme mit endlichen Automaten / Moore-Maschinen kann ganz hervorragend am Beispiel des Blutdruckmessgeräts erfolgen. Lernen die SuS SQL, so kann der Lehrer geeignete Tabellen zur Verfügung stellen, so dass die SuS medizinischen Fragestellungen auf den Grund gehen können wie den folgenden: Sind Jungen im Durchschnitt bei der Geburt größer als Mädchen? Erkranken Raucher öfter an Lungenkrebs als Nichtraucher? Leiden Übergewichtige öfter unter Diabetes als Normalgewichtige? Ein Beispiel für hierarchische Strukturierungen oder Codierung ist das Diagnoseklassifikationssystem ICD10. Setzt man im Unterricht UML-Diagramme ein, um die Strukturen im Gesundheitswesen und alle am Gesundheitssystem Beteiligte zu identifizieren (vgl. [St12]), ist zwar eine Recherche / Interviews in diesem Bereich der medizinischen Administration und Organisation notwendig, um erfolgreich sein zu können. Auf der anderen Seite umkleidet der Kontext aber für die SuS im Grunde nur das eigentliche Ziel, UML zu lernen.

In der zweiten Kategorie liefert die Medizin Vorlagen für Entwicklungen der Informatik. Beide Aspekte, hier auch die Medizin, müssen vertieft werden. **Beispiele:** Ist das RGB-Modell Unterrichtsthema, kann zugleich die Physiologie des Auges thematisiert werden, die zur Auswahl dieses Farbmodells führt. Ist das Unterrichtsthema „der lernende Roboter“, kann der Pawlowsche Versuch und der Aufbau eines Neurons Thema sein und anschließend auf sehr einfache Weise z.B. mit NXT-Robotern umgesetzt werden: Mit Hilfe von zwei Variablenwerten werden korrelierte Messwerte zweier Sensoren (z.B. Geräuschsensor und Berührsensor) gekoppelt. So kann das „angeborene“ Verhalten (Umdrehen bei Aufprall) auf den akustischen Reiz übertragen werden.

Die dritte Kategorie entspricht dem Motto „Informatik hilft!“. Der soziale Aspekt bildet hier den Ausgangspunkt einer Projektphase. Das Vorgehen und die Abgrenzung gegen die anderen beiden Kategorien soll zunächst an zwei Beispielen entwickelt werden, bevor eine Definition der Kategorie gegeben wird.

Beispiel 1: SuS der Sek 1 programmieren in scratch [sc13] eine Simulation des Blutzuckerkreislaufs. Die Objekte „Pizza/Essen“, „Muskel“, „Leber“ und „Fettzelle“ interagieren über Nachrichten miteinander. Der Blutzuckerwert (ein Variablenwert) erhöht sich nach dem Essen („Pizza“ wird angeklickt). Liegt er über einem bestimmten Schwellenwert, wird Insulin (Nachricht „Insulinausschüttung“) ausgeschüttet, was dafür sorgt, dass Glukose in Leber und Muskelzellen gespeichert wird (Änderung der Variablenwerte). Dadurch sinkt der Blutzuckerwert. Sind die Glukosespeicher von Leber und Muskeln gefüllt, kann auch in den Fettzellen Zucker in Fett umgewandelt werden und die Fettzellen vergrößern sich. Bei zu wenig Glukose im Blut durch Muskelbewegung („Muskel“ wird angeklickt) oder langen Essenspausen (Zeit), wird das Hormon Glukagon ausgeschüttet, das umgekehrt Glukose aus den Speichern von Leber und Muskeln ins Blut abgeben kann und den Blutzuckerwert so erhöht. (vgl. [Fa88], [mi13]). Die „Bühne“ in scratch kontrolliert ständig die Veränderung des Blutzuckerwertes, und bei zu hohem Zuckerwert oder zu geringem Zuckerwert werden die Objekte „Leber“, „Muskel“ und „Fettzelle“ informiert, die sich entsprechend der Physiologie verhalten. Soweit unterscheidet sich der Unterrichtsinhalt nicht von dem der ersten Kategorie. Die SuS implementieren eine Simulation in scratch und die Funktionalität des menschlichen Körpers liefert die Umkleidung der Aufgabe. Dem erwähnten Anspruch, den Kurs unter die Überschrift „Informatik hilft!“ zu stellen, wird so noch nicht Rechnung getragen. Wenn die Aufgabe aber lautet, den „zuckerkranken“ Patienten die tägliche schmerzhaftes Insulinspritze zu ersparen, dann ist eine „soziale“ Basis gegeben, von der aus die SuS das System des Blutzuckerkreislaufs recherchieren, nachbilden und in ihr System eine automatische implantierte Insulinpumpe einbauen müssen, die die Insulinabgabe automatisch regelt und die schmerzhaftes Spritze und Kontrolle des Zuckerspiegels ersetzt. Ausgehend von der unbefriedigenden Situation, der sich ein Diabetespatient ausgesetzt sieht (vielleicht gibt es jemanden in der Schulgemeinschaft), können die SuS durch die Anwendung informatischer Werkzeuge dem Patienten helfen.

Beispiel 2: Eine Tageszeitung [tz11] brachte kürzlich einen Bericht über Modellwohnungen mit elektronischen Sensor-Netzwerken für Senioren, ein Forschungsprojekt von Ärzten und Informatikern mit dem Ziel, dass ältere Menschen länger in den eigenen vier Wänden leben können. Wir gehen davon aus, dass die SuS in einer Unterrichtseinheit Algorithmik scratch kennengelernt haben. Jetzt werden sie mit dem Modellvorhaben konfrontieren und können ihre Kenntnisse der Algorithmik anwenden, um diesen Menschen das Leben sicherer und bequemer zu machen. Mit dem picoboard [pi13] in Kombination mit scratch können verschiedene Sensoren an den Rechner angeschlossen und die Sensorwerte im Programm verwendet werden (vgl. [MS11]). So können SuS z.B. akustische Signale und Texthinweise produzieren, wenn der Bewohner den Herd angeschaltet hat (Sensor „Schalter“), es Abend ist („Lichtsensor“ außen am Haus) und er zu Bett geht („Drucksensor“ im Bett). Viele andere Szenarien sind denkbar. Die SuS setzen ihre eigenen Ideen und Ziele mit zuvor erlernten informatischen Kenntnissen um. Die Ergebnisse der Aufgabe sind offen, vielfältig und von den eigenen Ideen der SuS geprägt. Sie sind in diesem Sinne kreativ (vgl. [Ro07]) und einer Binnendifferenzierung förderlich, da jede Idee erfolgreich, die Ideen aber unterschiedlich komplex sein können.

Zur Kategorie „Informatik hilft!“ gehören also Projekte, die das Ziel verfolgen, Menschen zu helfen. In einer *Projektphase* wenden die SuS zuvor erlernte informatische Strategien, Verfahrensweisen und Werkzeuge an. Die Problemstellung muss so offen sein, dass die SuS ihre eigenen Ideen kreativ umsetzen können. Der medizinische Kontext erfordert dabei zwingend eine Vertiefung in biologischen / medizinischen Kenntnissen, die integrativ behandelt werden können. Mit „Informatik hilft!“ ist also die Aufforderung gemeint: „Ihr SuS könnt mit euren Ideen Menschen helfen. Die Informatik liefert euch die Werkzeuge dazu. Strengt euren Kopf an!“ Dann ergibt sich auch der von Modrow [Mo10] zitierte „Sinnbezug“ der Informatik, und hier sieht die Autorin den Anreiz und Relevanzbereich für Mädchen.

Natürlich wird nicht in der gesamten Sekundarstufe 1 der Unterricht an diesen Inhalten ausgerichtet sein. Bei der Autorin ergab sich in der Praxis eine Verteilung wie Abbildung 2 zeigt.

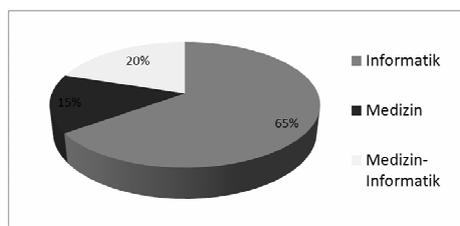


Abbildung 2: Verteilung der Inhalte Informatik, Medizin, Medizin-Informatik

Wir beschreiben im folgenden Kapitel drei „Informatik hilft!“-Projekte ausführlicher, die aus drei verschiedenen Inhaltsbereichen der Informatik der Sek 1 stammen.

3 Unterrichtsprojekte

3.1 Technische Informatik im Kontext „myoelektrische Armprothesen“

Informatische Voraussetzungen: logische Verknüpfungen, zusammengesetzte Schaltterme, Gatterdarstellung.

Medizinische Fragestellungen: Reizweiterleitung, Gegenspielerprinzip bei Muskeln.

Ziel: Für ein Unfallopfer soll eine Prothese so angefertigt werden, dass es seinen Beruf weiter ausüben kann.

Das Projekt selbst wird anhand einer konkreten Patientengeschichte entwickelt. Myoelektrische Armprothesen bilden einen medizinischen Kontext, der einerseits realitätsnah ist, andererseits zu sehr einfachen Lösungen, z.B. $x \square \neg y$, führt. Damit brauchen wir in der Sek 1 nicht auf das Basteln mit digitaler Elektronik zu verzichten. Mit Hilfe des Arduino-Boards [ar13] und einer Software, welche die SuS *anwenden* und der Lehrer in unserem Fall zuvor in Processing [pr13] erstellt hat, stecken die SuS Schaltnetze für myoelektrische Armprothesen mit digitaler Elektronik (Abbildung 3). Die Funktionalität der Prothese ergibt sich dabei allein durch die gesteckten Schaltnetze.

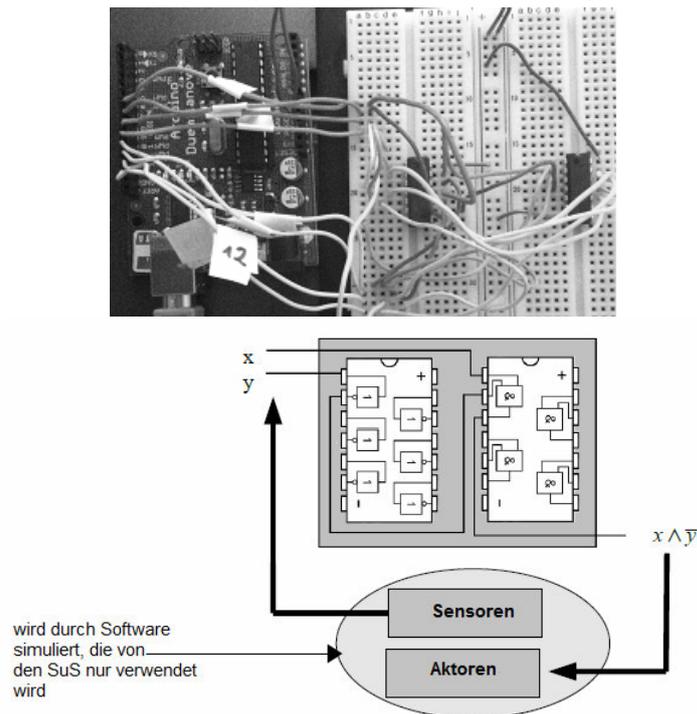


Abbildung 3: Schaltnetz für Armprothese

„Bei jeder Kontraktion des Muskels entsteht auf der Haut eine elektrische Spannung, die zum Steuern elektrisch angetriebener Prothesen genutzt wird.“ [ot13]. Individuell für jeden Patienten werden die elektrischen Signale, die noch willentlich erzeugt werden können, gemessen und in digitale Signale umgewandelt, die zur Steuerung der Motoren der Armprothese genutzt werden. Mit der verwendeten Software können die SuS mit der Maus vier Sensoren alternierend auf low oder high (Abbildung 4) setzen. Diese digitalen Signale werden via Arduino-Bord auf die Steckplatine übertragen. Hier können die SuS mit einfachen UND- und NICHT Bausteinen beispielsweise das Gegenspielerprinzip bei Muskeln stecken. Beispiel: zwei der Sensoren registrieren die elektrischen Signale von Bizeps und Trizeps. Erreicht den Bizeps ein elektrisches Signal, den Trizeps jedoch nicht, soll der Unterarm der Prothese gegen den Oberarm gebeugt werden. Die SuS verknüpfen also die beiden zugehörigen Sensorsignale mit einem Term der Form $x \square \neg y$ und leiten das Ergebnis zum Arduino-Board und damit zur Software zurück. Liegt an einem bestimmten Pin des Boards nun ein high-Signal, wird der Ellenbogen um 1 Grad gegen den Oberarm gebeugt. Durch eine hohe Abtastrate der Boardeingänge wird eine fließende Bewegung der Prothese simuliert. Ist das Schaltnetz dann der Aufgabe entsprechen gesteckt und mittels Software getestet, muss es (gedanklich) in die Prothese verlagert werden. Dort wird es über eine Batterie mit Spannung versorgt, die Eingänge werden von Ärzten mit den Sensoren an den Muskeln, die die myoelektrischen Signale aufnehmen, verschaltet und die Ausgänge des Schaltnetzes steuern Motoren, so dass die gesteckte Logik der SuS nicht schon in die Testsoftware integriert werden kann (vgl. [St13]).

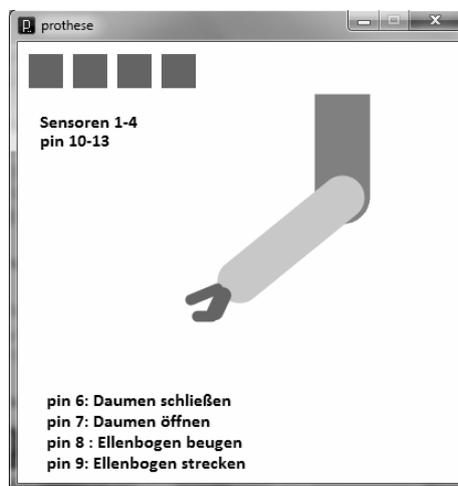


Abbildung 4: screenshot der Prothesen-Software

3.2 Informatik und Gesellschaft am Kontext elektronische Patientenakte (ePA)

Informatische Vorkenntnisse: unterschiedlich, z.B. Algorithmik, Kryptografie, Datenbanken, UML oder keine.

Medizinische Fragestellungen: Recherchen wie weiter unten beschrieben.

Ziel: Vorteile der ePA für Patienten ohne datenschutzrechtliche Nachteile nutzbar machen.

Die ePA ist eine digitale Sammlung aller behandlungsrelevanten Daten eines Patienten und enthält Dokumente wie Arztbriefe, Befunde, Diagnosen, Therapien und medizinische Bilddateien (z.B. Röntgenbilder). Damit soll in erster Linie die Kommunikation aller an der Gesundheitsversorgung Beteiligten verbessert werden, was dem Patienten z.B. Doppeluntersuchungen erspart oder Wechselwirkungen von Medikamenten, die bei verschiedenen Institutionen verschrieben wurden, verhindert. Im Notfall kann z.B. das Vorhandensein von Allergien einrichtungsübergreifend erkannt werden. Neben der Verbesserung der Kommunikation gilt die Kostensenkung als wichtiger Vorteil. Kritiker dagegen befürchten, dass die gesammelten Daten für andere Zwecke verfügbar gemacht werden. Die ePA wird aktuell diskutiert und betrifft früher oder später jeden Schüler. Fachleute wie z.B. Ärzte für Interviews lassen sich leicht in der Elternschaft finden.

Lässt man die Schülerinnen und Schüler zunächst im Internet recherchieren, um eine Definition der ePA zu finden, Vor- und Nachteile, Voraussetzungen und Risiken herauszuarbeiten und später in einer Art Talkshow (vgl. [Pe99]) über Sinn und Zweck der ePA zu diskutieren, ist immer wieder zu beobachten, dass sich die Diskussion auf Initiative einiger SuS in eine datenschutzrelevante Richtung verschiebt. Es ergeben sich oft schon von allein Fragestellungen wie folgt: Der Patient muss bestimmen dürfen, welche Daten er anderen Teilnehmern im Gesundheitssystem zur Verfügung stellen möchte, denn ein Zahnarzt muss nicht unbedingt die Diagnosen des Hautarztes kennen, höchstens die aktuelle Medikation (Stichpunkt: informationelle Gewaltenteilung). Was aber ist bei Bewusstlosigkeit? Hier wäre es von Vorteil, wenn ein Notarzt sich über Allergien informieren könnte auch ohne Zustimmung des Patienten (Stichpunkt: Recht auf informationelle Selbstbestimmung). Eine Apotheke darf Informationen über aktuelle Krankheiten nicht für Werbezwecke verwenden (Stichpunkt: Zweckentfremdungsverbot).

Diese Diskussion muss Konsequenzen haben. Die SuS sollen ihre informatischen Kenntnisse einsetzen, um die Vorteile der ePA für den Patienten teilweise so umzusetzen, dass datenschutzrechtlich keine Bedenken bestehen. Dieses kann auf verschiedene Weisen in Abhängigkeit von den Vorkenntnisse oder den Ideen der SuS geschehen. Beispiel: Die SuS könnten in einer weiteren Modellierungsphase, eventuell verbunden mit Interviews mit Ärzten, genauer eruieren, welche Daten im System ePA vorliegen. Ein Schülerergebnis dazu zeigt Abbildung 5. Die SuS könnten aber genauso gut in der Sprache BYOB [by13] die Möglichkeit der Verbindung zweier Programmierumgebungen auf zwei verschiedenen Rechnern nutzen (vgl. „mesh“-

Funktion) und sich um eine geschützte Datenübertragung mit Hilfe einfacher Prinzipien der Kryptografie bemühen. (Für Verschlüsselungen, die auf der Verschiebung des ascii-Zeichensatzes beruhen, bietet BYOB vorgegebene Funktionen). Haben die SuS bereits Erfahrungen mit SQL, könnten sie sich auch damit beschäftigen, wie die einzelnen Tabellen zu strukturieren sind, damit der Zahnarzt eben nicht die Diagnosen des Hautarztes sieht und der „informationellen Gewaltenteilung“ Rechnung getragen werden kann. Die Tabellen könnten auch so organisiert werden, dass behandlungsrelevante Daten anonymisiert verwaltet werden, damit sie auch der medizinischen Forschung zur Verfügung gestellt werden können.

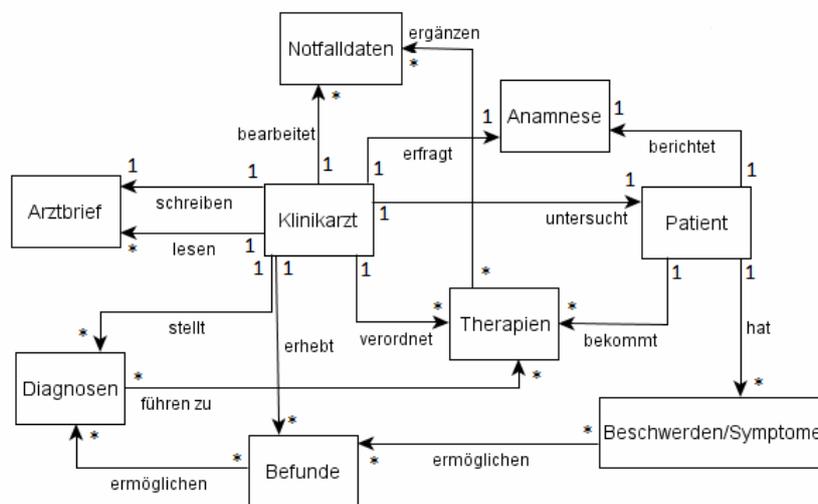


Abbildung 5 : Schülerergebnis Fachklassendiagramm Behandlung eines Patienten

3.3 Algorithmik im Kontext Bildbearbeitung / Biosignalverarbeitung

Informatische Vorkenntnisse: Algorithmik / scratch

Medizinische Fragestellungen: Einsatz, prinzipielle Funktionsweise und Unterschiede bildgebender Verfahren.

Ziel: Diagnoseunterstützung des Arztes.

Zunächst werden die bildgebenden Verfahren Ultraschall und Röntgenaufnahmen näher betrachtet. Beide Verfahrensprinzipien können von SuS der Sek 1 verstanden werden, es finden sich didaktisch aufbereitete Materialien dazu auch im Netz, siehe z.B. [wr13]. Neben eigenen Erfahrungen kennen die SuS das Röntgen meist auch im Zusammenhang mit Archäologie, Ultraschall oft aus der Biologie als Orientierungsprinzip von z.B. Fledermäusen. Folgende Projekte werden den „Informatik hilft!“-Kriterien gerecht:

Beispiel 1: In der Schwangerschaft werden Ultraschallaufnahmen des Embryos gemacht. Eine wichtige Untersuchung ist hierbei das Ausmessen der Größe des Embryos. Der Arzt markiert auf dem Ultraschallbild Kopfende und Rumpfende und die Software

berechnet automatisch den Abstand zwischen diesen Punkten. (Im Netz finden sich Filme dieser Untersuchungsmethode). Ein Ultraschallbild für die Bühne findet sich ebenfalls im Netz [wi13].

Lässt man die SuS dieses medizin-technische Gerät rekonstruieren, so ergeben sich die vielfältigsten Lösungsansätze. Man kann ein Objekt „Lineal“ erzeugen, mit den Pfeiltasten steuern und drehen und so die Größe ablesen. Man kann Markierungsobjekte zum Kopf- und Rumpfe mittels Maus oder Pfeiltasten bewegen und dort „Abdrücke hinterlassen“. Anschließend kann man mit scratch-Befehlen oder dem „Satz des Pythagoras“ den Abstand berechnen.

Beispiel 2: Röntgenaufnahmen sollte man sich als „Bühnenhintergrund“ am besten selbst zeichnen, wie Abbildung 6 (1) zeigt. Dann kann ein Programm geschrieben werden, das z.B. automatisch eine Bruchstelle identifiziert.



Abbildung 6 : scratch-Bühnen bei automatischer Röntgenbildauswertung (1), EKG-Auswertung (2) und Embryovermessung (3)

Hier gibt es ebenfalls vielfältige Lösungsansätze, was dem Kreativitätskriterium entspricht. Ein Schülerergebnis soll hier herausgestellt werden, weil es auch bei einer automatischen Auswertung von EKG-Signalen zur Identifikation eines drohenden Herzinfarktes o.ä. helfen kann. Die Schüler haben ein Objekt gebastelt, das umgeben ist von verschiedenfarbigen Punkten. Scratch liefert Befehle, mit denen man abfragen kann, ob eine bestimmte Farbe eine andere Farbe berührt. So kann sich das Objekt im Inneren des Knochens oder auch auf einer EKG-Linie bewegen. Die x- und y-Positionen der EKG-Kurve können beispielsweise abgefragt und so z.B. der Abstand zwischen Minimum und Maximum ermittelt werden. Auf ähnliche Weise kann mit dem Objekt eine schwarze Bruchstelle im weißen Knochen gefunden werden.

4 Fazit

Die Beispiele und deren Fülle sollen belegen, dass eine Kombination von Informatik und Medizin im gesamten Unterrichtszeitraum und quer durch alle informatischen Inhaltsbereiche der Sek 1 möglich ist und Raum für produktorientierten, kreativen und projektbetonten Unterricht unter dem Schlagwort: „Informatik hilft!“ bietet. Die notwendigen medizinischen Vorkenntnisse sind auch für Informatiklehrer ohne

biologische oder medizinische Ausbildung schnell zu erwerben. Nach Erfahrungen der Autorin sind Anzeichen zu erkennen, dass diese Form von Unterricht besonders Mädchen motiviert, Informatik in dieser Kombination zu wählen.

Literaturverzeichnis

- [ar13] URL: <http://www.arduino.cc/>, Zugriff: 31.1.2013
- [by13] URL: <http://byob.berkeley.edu/>, Zugriff: 31.1.2013
- [DKW11] Ira Diethelm, Jochen Koubek, Helmut Witten: IniK – Informatik im Kontext, Entwicklungen, Merkmale und Perspektiven, Zeitschrift LOG IN, Heft 168/170, 2011
- [Fa88] Adolf Faller: „Der Körper des Menschen“, Thieme-Verlag, 1988
- [LM02] Thomas Lehmann, Erdmuthe Meyer zu Bexten: „Handbuch der Medizinischen Informatik“, Hanser-Verlag, 2002
- [mi13] URL: <http://www.miomedi.de/gesundheit/krankheiten/diabetes/regulation-blutzucker/regulation-blutzucker.html>, Zugriff: 31.1.2013
- [Mo10] Eckart Modrow: „Informatik als technisches Fach“, Zeitschrift LOG IN, Heft 163/164, 2010
- [MS11] Eckart Modrow, Kerstin Strecker: „PuMa II“, Zeitschrift LOG IN, Heft Nr. 169/170, LOG IN-Verlag, 2011
- [ot13] URL: http://www.ottobock.de/cps/rde/xchg/ob_de_de/hs.xsl/5057.html, Zugriff: 31.1.2013
- [Pe99] Wilhelm H. Peterßen: Kleines Methoden-Lexikon, Oldenbourg Verlag, 1999
- [pi13] URL: <http://www.picocrocket.com/picoboard.html>, Zugriff: 31.1.2013
- [pr13] URL: <http://processing.org/>, Zugriff: 31.1.2013
- [Ro07] Ralf Romeike: „Kriterien kreativen Informatikunterrichts“, INFOS2007, GI-Lecture Notes in Informatics, 2007
- [St12] Kerstin Strecker: „Systemanalyse mit UML am Beispiel der Modellierung elektronischer Patientenakten“, in Marco Thomas, Michael Weigend (Hrsg.): „Ideen und Modelle, 5. Münsteraner Workshop zur Schulinformatik“, Books on Demand GmbH, Norderstedt, 2012
- [St13] Kerstin Strecker: „Myoelektrische Armprothesen und digitale Elektronik“, angenommen von der Zeitschrift LOG IN, voraussichtlich: Heft 174, 2013
- [SW06] Martina Schuster, Almut Sülzle, Gabriele Winker, Andrea Wolfram: „Neue Wege in Technik und Naturwissenschaften– Zum Berufswahlverhalten von Mädchen und jungen Frauen“, Baden-Württemberg / Wirtschaftsministerium, 2006, URN: urn:nbn:de:gbv:830-opus-1375, URL: <http://doku.b.tu-harburg.de/volltexte/2006/137/>
- [tz11] Hessisch-Niedersächsische Allgemeine vom 1.8.2011
- [wi13] URL: http://commons.wikimedia.org/wiki/File:Medecine_Echographie.jpg, Zugriff: 31.1.2013
- [wr13] URL: <http://www.werkstatt-roentgen.de/>, Zugriff: 31.1.2013

Das 3x3-Schema zur Reflexion kontextorientierter Informatik-Unterrichtsgestaltung

Malte Buchholz

AG Didaktik der Informatik
Freie Universität Berlin
Königin-Luise-Str. 24-26
14195 Berlin
m.buchholz@fu-berlin.de

Abstract: Die Organisation strukturierter Reflexion innerhalb von Unterrichtsentwicklungs-Projekten hilft nicht nur das Produkt, sondern auch den Prozess der Unterrichtsgestaltung zu verbessern. In Anlehnung an bestehende Konzepte aus der *Aktionsforschung*, dem *Design Based Research*, *Didaktischer Entwicklungsforschung* und der praktischen Erfahrungen innerhalb des Berliner „Informatik-im-Kontext (IniK)-Sets“ wird in diesem Beitrag ein theorie-gebundenes und praxisorientiertes Konzept entwickelt, das erlaubt, kontext-orientierte Informatik-Unterrichtsgestaltung strukturiert zu reflektieren. Das Konzept richtet sich vor allem an IniK-Sets, sowie Ausbilder der ersten und zweiten Phase der Lehramtsausbildung.

1 Einleitung

1.1 Ziele des Reflexionskonzepts

Seit 2008 arbeitet das Berliner IniK-Set an der Gestaltung von Unterrichtseinheiten nach dem Unterrichtskonzept „Informatik im Kontext“. Während die Produkte des Sets bereits mehrfach Gegenstand von Veröffentlichungen waren, ist der Prozess der Unterrichtsgestaltung durch das Set nicht ausreichend dokumentiert. Die Gestaltung von kontextorientierten Unterrichtseinheiten ist erfahrungsgemäß herausfordernd und zeitaufwändig. Im Berliner IniK-Set gehören erfolgreiche Planungsvorhaben ebenso dazu wie gescheiterte. Durch wechselnde Akteure und meist nur unstrukturierte mündliche ad-hoc Weitergabe gehen viele Erfahrungen in Bezug auf den Planungsprozess verloren. Dokumentiert wird meist nur das finale Produkt. Hieraus ergibt sich das erste Ziel des hier vorgestellten Schemas zur Reflexion von IniK-Einheiten:

(1) *Eine strukturierte und dokumentierte Reflexion erlaubt es, den Gestaltungsprozess für Dritte transparent zu machen:* Erkenntnisse über Herausforderungen und Chancen, sowie Organisation des Gestaltungsprozesses können von anderen Arbeitsgruppen/ Akteuren genutzt werden, um deren eigenen Prozess erfolgreich zu gestalten.

Das Berliner Set hat den Anspruch, qualitativ hochwertige Unterrichtseinheiten zu produzieren. Die Reflexion der Unterrichtsgestaltung ist dabei ein unverzichtbares Mittel, um die Qualität der Entwürfe zu gewährleisten. Geht man davon aus, dass tief-gehende Reflexion entscheidend dazu beitragen kann, die Unterrichtseinheiten zu verbessern, ergibt sich daraus das zweite Ziel des hier vorgestellten Schemas:

(2) *Eine in den Entstehungsprozess eingeflochtene Reflexion erlaubt es, die Qualität der Unterrichtseinheiten zu erhöhen:* Die Gestaltung der Reflexion muss sich an den Grundlagen des Unterrichtskonzepts orientieren. Sie darf nicht auf nachträglicher Einsicht nach Fertigstellung des Produkts beruhen – sie muss in den Entstehungsprozesses integriert werden.

Wie an den Formulierungen der Ziele (1) und (2) deutlich wird, legt das hier vorgestellte Reflexionskonzept einen Schwerpunkt auf den Gestaltungsvorgang, nicht auf das entstehende Produkt. Das Konzept ist somit prozess-, nicht produktorientiert.

1.2 Methodisches Vorgehen

Zur Konzeption der Reflexion wurde folgendes Vorgehen gewählt: Es wird zunächst dargestellt, was die wesentlichen Grundzüge des Unterrichtskonzepts „Informatik im Kontext“ (IniK) sind und wie das Berliner IniK-Set organisiert ist. Aus den Grundzügen von IniK und der Organisation des Sets werden konkrete Rückschlüsse für die Ausgestaltung der Reflexion gezogen. Darauf folgend werden bereits vorhandene Ansätze aus der Forschung untersucht, um anschließend aus ihnen konkrete Maßnahmen für die Gestaltung der Reflexion in überschulischen Projekten zu gewinnen. Es werden vor allem Vorschläge aus der *Aktionsforschung*, dem *Design Based Research* und der *Didaktischen Entwicklungsforschung* betrachtet. In einem dritten Schritt werden die theoretischen Grundlagen und die konkreten Rahmenbedingungen des IniK-Projekts miteinander verknüpft und ein Reflexionskonzept (das „3x3-Schema zur Reflexion kontextorientierter Unterrichtsgestaltung“) abgeleitet, das es ermöglicht die in Kap. 1.1 definierten Ziele zu erreichen.

2 Was ist „Informatik im Kontext“(InIK)?

2.1 Die Grundzüge von IniK

IniK ist ein Unterrichtskonzept für den Informatikunterricht, das 2008 ins Leben gerufen wurde. Vorbilder für das Konzept waren die Kontextprojekte aus der Physik (PiKo), der Chemie (ChiK) und der Biologie (biK). Im Gegensatz zu den naturwissenschaftlichen Kontextprojekten ist IniK ein „Graswurzelprojekt“, was bedeutet, dass IniK weitestgehend ohne finanzielle Förderungen besteht – in Berlin wird das Projekt jedoch durch die Bereitstellung von Abordnungsstunden durch die Senatsverwaltung unterstützt [KSW09]. Das Konzept ist als Gegenentwurf zu dem gängigen Bild von Informatikunterricht als ‚Anwenderschulung‘ in der Sekundarstufe I und ‚Programmierkurs‘ in der Sekundarstufe II angelegt [EP10a, S. 115]. Es beruht im Wesentlichen auf drei Säulen [KSW09]:

(1) *Orientierung an Kontexten:* Die Grundideen der Informatik werden im IniK-Unterricht in Kontexte eingebettet unterrichtet. Ein Kontext ist „[eine] Menge von lebensweltlichen Themen bzw. Fragestellungen, die von den Schülerinnen und Schüler als zusammenhängend geordnet werden und die dadurch sinnstiftend auf deren Handlungen wirken“ [KSW09]. Der Lebensweltbezug steht somit im Zentrum von IniK-Einheiten. Dadurch wird sich erhofft, dass ein realistisches Bild der Informatik in der Schule vermittelt werden kann: Die SchülerInnen sollen die Grundideen der Informatik als relevantes Wissen und Können für ihre eigene Lebenswelt – und nicht als losgelöstes Expertenwissen – kennenlernen.

(2) *Orientierung an Standards für Informatik in der Schule*: IniK-Unterrichtseinheiten sollen einen Kompetenzzuwachs fördern, der SchülerInnen ermöglicht, gegenwärtigen und zukünftigen gesellschaftlichen Anforderungen zu genügen. Als bundesweites Projekt orientiert sich IniK dabei an den von der Gesellschaft für Informatik (GI) veröffentlichten Mindeststandards für das Fach Informatik. Die Standards sind das Produkt einer Kooperation von Akteuren aus Theorie und Praxis und treffen – trotz (oder wegen) eines lebhaften Diskurses – auf eine breite Zustimmung innerhalb der Fachgemeinschaft [GI08]. Bei der Entwicklung von IniK-Einheiten müssen diese Standards durch Konkretisierung des Planenden jedoch mit Leben gefüllt werden: Ein ‚bloßes Ankreuzen‘ der zu erreichenden Standards ist in jedem Fall zu vermeiden [EP10b].

(3) *Methodenvielfalt*: IniK fußt auf einem konstruktivistischen Lehr-Lern-Verständnis. Dementsprechend sollen IniK-Einheiten möglichst hohe Anteile von Methoden besitzen, die die Partizipation der SchülerInnen erlauben. Dabei gilt weiterhin die generelle Erkenntnis, dass die Methode zum Inhalt passen muss und die Methode zu bevorzugen ist, die den SchülerInnen am besten ermöglicht den angestrebten Kompetenzzuwachs anzubahnen.

2.2 Organisation und Arbeitsweise des Berliner IniK-Sets

Das Berliner IniK-Set wurde 2008 gegründet. Es besteht aus LehrerInnen, ForscherInnen/HochschullehrerInnen der Arbeitsgruppe ‚Didaktik der Informatik‘ der FU-Berlin und (teils ehemaligen) Informatik-Seminarleitern. Das Set wird durch die Senatsverwaltung für Bildung, Jugend und Wissenschaft durch zwei Abordnungsstunden unterstützt. Aus der Zusammensetzung des Sets ergeben sich auch die Zielrichtungen des Projekts: Erstellung von konkreten Unterrichtseinheiten, Stärkung der ersten und zweiten Phase der Lehramtsausbildung und Erkenntnisgewinnung innerhalb der fachdidaktischen Forschung. Der Fokus des hier entwickelten Reflexionskonzeptes liegt auf der kooperativen Erstellung von Unterrichtseinheiten. Der typische Ablauf bei der Erstellung von Unterrichtseinheiten ist wie folgt: 2-3 Akteure bringen eine Idee in das Set ein, stellen diese in einer Gruppensitzung vor, planen den Unterricht, führen ihn mehrfach durch und ändern entsprechend der im Unterricht gemachten Erfahrungen die Unterrichtsgestaltung. Wenn die Akteure die Unterrichtseinheit für ‚reif‘ halten, wird sie auf der Webseite des Projekts, bzw. als gedruckte Handreichung veröffentlicht. Bei Ideen, die von Seiten der Hochschule initiiert werden, wird die Unterrichtseinheit gemeinsam mit Studierenden zusätzlich im Schülerlabor Mathematik/Informatik (MI.Lab) der FU-Berlin durchgeführt und analysiert. Die Ausgestaltung der Unterrichtseinheiten wird regelmäßig innerhalb von Gruppensitzungen (ca. 4-5 pro Jahr) diskutiert. Die Unterrichtsgestaltung ist somit ein zyklischer Entwicklungsprozess, der Planungsphasen, Erprobungsphasen und Diskussionsphasen beinhaltet. Zusätzlich werden die Unterrichtseinheiten bundesweit auf Lehrerfortbildungen vorgestellt.

3 Theoretische Grundlagen des Reflexionskonzepts

Innerhalb der erziehungswissenschaftlichen und didaktischen Forschung existieren Konzepte, wie Unterrichtsentwicklung und -forschung innerhalb von überschulischen Projekten organisiert werden kann. Die hier vorgestellten Forschungsansätze haben gemeinsam, dass sie auf eine Verbindung von Theorie und Praxis innerhalb zyklischer Unter-

richtsentwicklung (bzw. Schulentwicklung) abzielen. Die drei Forschungsansätze *Aktionsforschung*, *Design Based Research* und *Didaktische Entwicklungsforschung* wurden als Grundlage für das Reflexionskonzept ausgewählt, weil sie direkt oder indirekt Aussagen über die Gestaltung von Reflexionsphasen innerhalb von Entwicklungsprojekten treffen. Dort wo in ihnen Verbindungen zur Reflexion hergestellt werden, beziehen sich die Ansätze auf die grundlegenden Arbeiten von Schön [Sch83]. In Kap. 4.1 werden die hier in aller Kürze erörterten theoretischen Grundlagen mit den Rahmenbedingungen des Berliner IniK-Sets verknüpft.

(1) *Aktionsforschung*: Altrichter und Posch stellen fest, dass es „im Lehrerberuf keine Tradition der Darstellung und Kommunikation des beruflichen Wissens, der [...] Erfahrungen und des [...] Nachdenkens“ gibt. Dies führt dazu, dass die Reflexion über die Praxis vernachlässigt wird [AP08, S.260]. Sie merken an, dass eine entsprechende Berufsorganisation notwendig sei, um die Praxis zu reflektieren und weiterzuentwickeln. Eine entsprechende Organisation könnte einen Beitrag zur eigenen, aber auch zur Professionalisierung Dritter leisten [ebd., S.278]. Als konkrete Möglichkeiten der Kommunikationsgestaltung werden folgende Mittel vorgeschlagen. (a) *Schriftliche Darstellungen*: können die Kommunikation stärken und sind Voraussetzung für Kritik und Anregungen durch Dritte. (b) *Kooperative Arbeitsformen*: Altrichter und Posch sehen das gegenseitige Besuchen von Unterricht, Lehrerfortbildungen und Gruppengespräche als Hilfsmittel der Reflexion. Dabei sollen Stärken und Schwächen identifiziert und besprochen werden, wie die derzeitige Praxis verbessert werden kann. (c) *Reflexion und Aktion als Kreislauf*: Die Entwicklung von unterrichtlicher Praxis ist kein sequentieller Prozess. Reflexion und Aktion bedingen sich gegenseitig und müssen deshalb als iterativer Prozess gestaltet werden.

Innerhalb der Didaktik der Chemie wird Aktionsforschung genutzt, um das Curriculum von „Chemie im Kontext“ zu entwickeln. Eilks und Ralle merken an, dass eine kooperative Unterrichtsgestaltung eine gemeinsame Reflexion gewährleistet. Als Grundlage der Reflexion und Evaluation des Gestaltungsprozesses nennen sie (d) *Beobachtungsberichte*, (e) *Erfahrungsberichte*, (f) eine *systematische Befragung* der „Praktiker“ durch die „Forscher“, (g) *dokumentierte Gruppendiskussionen* und (h) ausgefüllte *Schülermaterialien*. Auch Eilks und Ralle betonen, dass die Entwicklungsarbeit in einem iterativen Prozess gestaltet werden sollte [ER02].

(2) *Design Based Research (DBR)*: DBR ist ein Forschungsansatz, der dem Umstand gerecht wird, dass Unterricht in seiner Komplexität mit rein quantitativer empirischer Analyse kaum zu fassen ist. DBR konzentriert sich auf die Implementation von Interventionen, die darauf ausgerichtet sind, die schulische Praxis zu verbessern und gleichzeitig Theorie zu generieren [Br92]. DBR-Interventionen sind darauf ausgerichtet die Praxis zu innovieren. Reinmann schlägt auch hier ein iteratives Vorgehen vor, in dem Phasen des Designs, der Evaluation und des Re-Designs sich abwechseln. So angelegte Entwicklungsprozesse seien einerseits geeignet, um die Praxis zu verbessern – sie liefern einen „praktischen Output“. Andererseits können so geartete Prozesse Erkenntnisse über Lehr-Lern-Theorien und – dies ist für das hier entwickelte Reflexionskonzept entscheidend – auch Wissen über den Gestaltungsprozess hervorbringen („theo-retischer Output“). Reinmann betont weiterhin, dass einer Kooperation von Praktikern und Wissenschaftlern eine hohe Bedeutung zukommt. Bewertungskriterien für DBR-Prozesse seien: (a) Neuheit, (b) Nützlichkeit und (c) nachhaltige Innovation [Re05].

(3) *Didaktische Entwicklungsforschung*: Die Didaktische Entwicklungsforschung ist ein Forschungsansatz, der die Gestaltung von Lehr-Lern-Prozessen in den Mittelpunkt des Erkenntnisinteresses setzt. Sie zielt darauf ab das Theorie-Praxis-Problem in der Lehr-Profession zu überwinden [Ei10]. Tulodziecki und Herzig schlagen vor, an das Alltagshandeln folgende Fragen zu richten: (a) „Sind die Annahmen zu den Lernvoraussetzungen [der SchülerInnen] angemessen? (b) „Sind die Zielvorstellungen gerechtfertigt?“ (c) „Sind die [Lehr-Lern-Aktivitäten] geeignet, zur Erreichung der Ziele zu führen?“ [TH98, S.5]. Analog dazu, sollten die erzielten Ergebnisse (also das unterrichtliche Handeln und die dabei entstandenen Produkte) anhand folgender Fragen beurteilt werden: (a') „Wurden die Lernvoraussetzungen im Rahmen des Konzepts angemessen eingeschätzt?“ (b') „[...] Wurden die Lernenden in der angestrebten Weise aktiv?“ (c') „[...] Wie sind die erreichten Lernergebnisse im Aspekt der Zielvorstellungen zu beurteilen?“ [ebd., S.15]

Als Methoden, die eine Evaluation (und – wenn man annimmt dass Evaluation und Reflexion zwei Seiten einer Medaille sind – auch eine Reflexion) unterstützen nennt Einsiedler: (a) Unterrichtstagebuch, (b) Tiefeninterviews und (c) Portfolio. Hierbei sei besonders das Unterrichtstagebuch geeignet, um Erkenntnisse über den Gestaltungsprozess zu erlangen [Ei10].

4 Das 3x3-Schema zur Reflexion von IniK-Unterrichtsgestaltung

4.1 Rückschlüsse aus Praxis und Theorie für die Gestaltung des Reflexionskonzepts

Die drei Säulen des Unterrichtskonzepts „IniK“ bieten konkrete Anhaltspunkte dafür, wie die Reflexion kontextorientierter Unterrichtseinheiten gestaltet werden muss: Die Orientierung an Kontexten verlangt nach einer Reflexion über die Wahl des Kontexts. Neben der Frage, ob der Kontext „geeignet“ ist, müssen Einschätzungen und Erfahrungen über typische Lernvoraussetzungen und Präkonzepte von SchülerInnen reflektiert werden. Die Orientierung an Bildungsstandards gibt eine Zielsetzung vor: Der im Rahmen der GI-Standards angestrebte Kompetenzzuwachs durch die SchülerInnen muss im Mittelpunkt der Unterrichtsgestaltung stehen. Diese Zielsetzung betrifft auch die dritte Säule des IniK-Konzepts: Die methodische Vielfalt ist kein Selbstzweck, sondern muss dabei helfen, die definierten Ziele effektiv zu erreichen. Die von IniK gebotenen Anhaltspunkte für die Gestaltung des Reflexionskonzepts zeigen eine Deckung mit den von Tulodziecki und Herzig (s.o.) vorgeschlagenen Fragen an das Alltagshandeln von Praktikern. Aus den praktischen Voraussetzungen von IniK und dem theoretischen Konzept der Didaktischen Entwicklungsforschung können drei Fragen abgeleitet werden, die im Mittelpunkt des hier vorgestellten Reflexionskonzepts stehen:

- (1) Welche Lernvoraussetzungen müssen in Hinsicht auf den Kontext bei der Unterrichtsgestaltung beachtet werden?
- (2) Wie sieht eine angemessene Kompetenzentwicklung aus, die darauf abzielt, den SchülerInnen die Erreichung der gewählten Standards zu ermöglichen?
- (3) Wie müssen Lehr-Lern-Prozesse gestaltet werden, die eine solche Kompetenzentwicklung erlauben?

Wie dargestellt wurde, schlagen die drei vorgestellten theoretischen Konzepte vor, den Gestaltungsprozess iterativ zu gestalten, d.h., dass Phasen der Planung, der Durchführung und der Analyse/Reflexion zyklisch durchlaufen werden. Für die Gestaltung der

Reflexion bedeutet dies, dass sie kein einmaliger Prozess ist, sondern vielmehr in die iterative Entwicklungsarbeit integriert werden muss. Geeignete Stellen, an denen die Reflexion – der Erfahrung nach – eingebunden werden kann sind folgende: (1) *Reflexion während der Planung*, (2) *Reflexion nach der Durchführung* und (3) *Reflexion nach der Dissemination*. An jeder der drei Stellen soll die Unterrichts-gestaltung anhand der drei oben entwickelten Fragen reflektiert werden. Aus den Forschungsansätzen wurden verschiedene Vorschläge abgeleitet, mit welchen Mitteln und auf welche Art und Weise die Reflexion durchgeführt werden kann. Je nach Fortschritt der Unterrichtsgestaltung sollten sich Vorgehensweise und Mittel unterscheiden – die drei Fragen bleiben jedoch stets Grundlage der Reflexion. In Folge wird vorgestellt, wie die Reflexion innerhalb der Phasen des Entwicklungsprozesses jeweils umgesetzt werden kann.

4.2 Reflexion während der Planung

Der erste Schritt der Reflexion sollte noch vor der ersten Umsetzung der Unterrichtseinheit im Klassenraum erfolgen. Dadurch können Zielsetzungen geschärft und erste Entwurfsentscheidungen kritisch hinterfragt werden. Da – besonders im Anfangsstadium – während der Planung kaum schriftliche Produkte existieren, sollte die erste Reflexion als kommunikativer Gruppenprozess angelegt werden (s.o. Vorschlag ,1g‘). Die Planenden stellen ihre Unterrichtsidee während eines Set-Treffens vor und versuchen, die drei Fragen des Reflexionskonzepts zu beantworten. Um die Flüchtigkeit des Diskussionsprozesses zu umgehen, wird die Diskussion dokumentiert. Die Verschriftlichung sollte sich dabei an den drei vorgegeben Leitfragen orientieren. Falls die Planenden dem zustimmen, kann zusätzlich ein an den drei Fragen orientiertes, systematisch gestaltetes Interview zwischen WissenschaftlerInnen und Planenden durchgeführt werden (s.o. Vorschlag ,1f‘ & ,3b‘). In der Reflexion während der Planung sollten die drei Fragen unter der Beachtung folgender Vorschläge beantwortet werden:

Zu Frage (1) *Lernvoraussetzungen*: Die Frage nach den Lernvoraussetzungen ist einerseits eine Frage nach der Güte des Kontexts, andererseits eine Frage nach Präkonzepten von SchülerInnen. Ob ein Kontext angemessen ist, oder verworfen werden sollte, muss unbedingt während der Planung identifiziert werden. Dabei können v.a. die von [DD11] und [DKW11] entwickelten Kriterien für die Auswahl eines Kontexts hilfreich sein. Die Gütekriterien für die Auswahl eines Kontexts sind nicht als Ausschlusskriterien anzusehen – sie helfen jedoch, die Qualität der zu gestaltenden Unterrichtseinheit zu erhöhen. Ist die Diskussion des Kontexts erfolgt, sollten Präkonzepte von SchülerInnen in Bezug auf den Kontext in den Blick genommen werden. Es hat sich in der Praxis als fruchtbar erwiesen, über Wissen und Können, sowie Vorstellungen von SchülerInnen begründet zu spekulieren und die Gestaltung der Unterrichtseinheit entsprechend anzupassen.

Zu Frage (2) *Zielorientierung*: Der konstruktive Umgang mit den Bildungsstandards-Informatik der GI ermöglicht, die Ausrichtung der Unterrichtseinheiten auf eine ausgewählte Zielsetzung zu fokussieren. Es hat sich als nützlich erwiesen, nicht möglichst viele Standards innerhalb einer IniK-Reihe erreichen zu wollen – die Gestaltung der Einheit sollte vielmehr auf wenige Standards abzielen, z.B. die Standards nur eines Inhalts- und eines Prozessbereichs. Innerhalb der Gruppe sollte diskutiert werden, wie ein Standard in den Bezug auf den Kontext, die Lernaktivitäten und die verwendeten Operationsobjekte konkretisiert werden kann. Anschließend kann die vom Planenden entwickelte Unterrichtsidee entsprechend mit dem Blick auf die geplante Kompetenzentwick-

lung des Lernenden hinterfragt werden. Die Zielkonkretisierung bildet weiterhin die Grundlage für die Analyse und Reflexion nach der Unterrichtsdurchführung. Im Zuge der Zielkonkretisierung sollte auch festgehalten werden, anhand welcher Indikatoren man den Kompetenzzuwachs bei den SchülerInnen beobachten, bzw. messen könnte. Es gilt zu beachten, dass die GI-Standards nicht dogmatisch zu verstehen sind. Es wurde explizit darauf hingewiesen, dass IniK ein „proof of concept“ für die Bildungsstandards darstellt [KSW09]. Es können also auch Kompetenzentwicklungen angestrebt werden, die von den Unterrichtsgestaltenden als wichtig angesehen werden. Die Reflexion in der Gruppe bietet also neben der Möglichkeit die Qualität der eigentlichen Unterrichtseinheit zu fördern auch die Chance, die theoretischen Grundlagen von IniK weiterzuentwickeln.

Zu Frage (3) *Lehr-Lern-Prozesse*: Die Gestaltung der Lehr-Lern-Prozesse sollte anhand von zwei grundlegenden Kriterien reflektiert werden. Zum einen gilt es, die von IniK geforderte Methodenvielfalt zu erfüllen. Dabei ist vor allem die Partizipation der SchülerInnen am unterrichtlichen Handeln anzustreben. Zum anderen sind die Lehr-Lern-Prozesse an der Zielorientierung auszurichten. Die einzelnen Phasen der Unterrichtseinheit sollten auf Dienlichkeit für die angestrebte Kompetenzentwicklung diskutiert werden. Hier ist vor allem das Abwägen und das Suchen (und Festhalten) von Entwurfsentscheidungen, sowie eine klare Vorstellung darüber, welche Aktivitäten der Lernende durchführt, wichtig. Die Frage nach den Lehr-Lern-Prozessen betrifft – und das ist eine Besonderheit des Fachs Informatik – auch die eingesetzten Informatik-Operationsobjekte (Programme, Programmiersprachen, Hardware, etc.). Genau wie die Entscheidungen für bestimmte Unterrichtsmethoden sollte auch die Entscheidung für ein Operationsobjekt unter Berücksichtigung der zu erreichenden Ziele (und nicht etwa dogmatisch, z.B.: Was ist die ‚beste‘ Programmiersprache? Was ist das ‚beste‘ Betriebssystem? usw.) diskutiert und dokumentiert werden.

4.3 Reflexion nach der Durchführung

Eine zentrale Rolle innerhalb des Reflexionskonzepts spielt die Reflexion nach der (u.U. mehrfachen) Durchführung der Unterrichtseinheit. Diese Form der Reflexion bedarf jedoch weiterer Instrumente, um der Komplexität von unterrichtlichem Handeln gerecht zu werden. Der Unterricht kann zu einem gewissen Grad (vor allem von erfahrenen Kollegen) ad-hoc analysiert werden. Um eine strukturierte Reflexion zu ermöglichen, werden jedoch folgende Instrumente vorgeschlagen, die aus den oben vorgestellten Forschungsansätzen abgeleitet wurden: (1) Das gegenseitige Besuchen und Beobachten von Unterricht (s.o. Vorschlag ‚1b‘ & ‚1d‘) und (2) die Erstellung eines Unterrichtstagebuchs (s.o. Vorschlag ‚3a‘ & ‚1e‘). Die Ergebnisse der Beobachtung und der eigenen Erfahrungen in Form des Unterrichtstagebuchs können anschließend genutzt werden, um innerhalb der Projektgruppe in die erneute Diskussion einzutreten:

Zu Frage (1) *Lernvoraussetzungen*: Da die Frage nach der Eignung des Kontexts bereits während der Planung erörtert wurde, sollte der Kontext an sich nicht mehr zur Diskussion stehen. Deshalb wird vorgeschlagen vor allem Präkonzepte der SchülerInnen in Bezug auf den Kontext in den Fokus der Reflexion zu rücken. Es gilt zu überprüfen, ob die getroffenen Annahmen über Vorstellungen der SchülerInnen vollständig und korrekt waren.

Zu Frage (2) *Zielorientierung*: In Bezug auf die Zielorientierung soll der Prozess der Kompetenzentwicklung der Lernenden in den Blick genommen werden. Während der Reflexion in der Planungsphase wurden bereits Indikatoren identifiziert, die eine Beobachtung, bzw. Messung der Zielerreichung ermöglichen. Dabei sollten auch (s.o. Vorschlag ‚1h‘) Schülerprodukte analysiert werden. Die Gedanken über die Zielerreichung sollten im Idealfall mit dem Beobachtenden besprochen und dokumentiert werden, mindestens aber im Unterrichtstagebuch festgehalten werden. Bei einer nichterfolgreichen (aber auch bei einer erfolgreichen) Zielerreichung sollten Spekulationen über Erfolgsfaktoren festgehalten werden, um diese anschließend in der Projektgruppe diskutieren zu können.

Zu Frage (3) *Lehr-Lern-Prozesse*: Die Reflexion der Lehr-Lern-Prozesse sollte darauf abzielen, die Frage zu beantworten, ob die Lernenden in der Art und Weise aktiv geworden sind, wie es in der Planung angelegt wurde. Wie konkretes Handeln von SchülerInnen aussieht, vor allem bei komplexen Methoden wie z.B. Rollenspielen, kann während der Erstplanung nur ansatzweise antizipiert werden. Die Reflexion über Handlungen im Unterricht gibt deswegen wertvolle Ansatzpunkte für eine Überarbeitung der Planung. Spezifisch für die Informatik sollte auch beobachtet werden, wie die SchülerInnen die angebotenen Operationsobjekte nutzen. Im Anschluss sollte reflektiert werden, ob das Objekt entscheidend für die Zielerreichung war, oder ob es umgestaltet werden muss.

4.4 Reflexion nach der Dissemination

Wann (und ob) eine Unterrichtsgestaltung abgeschlossen ist, ist schwer zu entscheiden. Eine Kultur der Verschriftlichung und Veröffentlichung der Unterrichtseinheiten kann jedoch nicht nur Transparenz herstellen, sondern auch die Qualität der Unterrichtsgestaltung erhöhen. Während der letzten Reflexionsphase gilt es, Meinungen innerhalb der Gemeinschaft der Schulinformatik zu sammeln. Anschließend sollten die so gewonnenen Erkenntnisse gesammelt und diskutiert werden und die Unterrichtsgestaltung entsprechend angepasst oder ergänzt werden. Bei der Sammlung der Erfahrungen, die andere LehrerInnen beim Durchführen der veröffentlichten Einheit gemacht haben, kann auf eine (reduzierte) Form des Unterrichtstagebuchs zurückgegriffen werden. Auch hier ist es entscheidend, einen attraktiven und handlungs-auffordernden Vordruck bereitzustellen, in dem die drei Leitfragen im Zentrum stehen. Dadurch kann gewährleistet werden, dass die von Dritten gemachten Erfahrungen innerhalb des vorgestellten Rahmens reflektiert werden. Eine zweite Möglichkeit, die vom Berliner IniK-Set bereits genutzt wird, ist es, die Unterrichtseinheiten in Lehrerfortbildungen zur Diskussion zu stellen.

Es wird vorgeschlagen, den Gestaltungsprozess einer kontextorientierten Unterrichtseinheit mit einer dokumentierten Abschlussdiskussion innerhalb der Projektgruppe zu beenden. Hierbei sollten erstmals nicht die Leitfragen im Mittelpunkt stehen, sondern der gesamte Gestaltungsprozess reflektiert werden. Ziel ist es dabei, als wichtig wahrgenommene Entwurfsentscheidungen für Dritte transparent zu machen. Dabei kann auf die – vorher aufbereiteten – Materialien, die während des Reflexionsprozesses entstanden sind, zurückgegriffen werden. Es wird vorgeschlagen diese Abschlussreflexion an den DBR-Bewertungskriterien (s.o., Vorschlag ‚2abc‘) *Neuheit*, *Nützlichkeit* und *nachhaltige Innovation* auszurichten. Die so gewonnenen Erfahrungen und Erkenntnisse sollten festgehalten und für das nächste Entwicklungsprojekt genutzt werden. Somit kann neben der

Transparenz auch die Qualität der zukünftigen Arbeitsweise innerhalb des IniK-Sets gefördert werden.

5 Fazit und Ausblick

Das hier entwickelte Reflexionskonzept zielt darauf ab, die oben definierten Ziele *Qualität* und *Transparenz* zu erreichen. Um eine strukturierte und zielorientierte Reflexion zu ermöglichen, wurde das „3x3-Schema“ entwickelt, das eine Konzentration auf drei Leitfragen innerhalb der drei Phasen der Unterrichtsgestaltung vorsieht. Das Konzept wurde sowohl auf Basis von Erkenntnissen aus der Bildungsforschung, als auch auf Basis von praktischen Erfahrungen innerhalb des Berliner IniK-Sets entworfen. Die Zielgruppen des Reflexionskonzepts sind Ausbilder der ersten und zweiten Phase der Lehramtsausbildung, sowie Unterrichtsentwicklungs-Projektgruppen. Die Reflexion stützt sich vor allem auf die Instrumente *dokumentierte Gruppendiskussion*, *Unterrichtstagebuch* und *gegenseitiges Beobachten von Unterricht*. Es wurden Vorschläge gemacht, wie die Akteure aus Wissenschaft und Schule innerhalb des Reflexionskonzepts agieren können. Bei der Umsetzung des Konzepts in die Praxis muss vor allem darauf geachtet werden, dass die Akteure nicht übermäßig belastet werden. Aus diesem Grund ist besonders das Unterrichtstagebuch vorab so zu gestalten, dass es leicht auszufüllen ist und dem/der Unterrichtplanenden einen möglichst großen Nutzen bringt. Auf lange Sicht könnte aus der Erfahrungen mit dem Reflexionskonzept ein Leitfaden entstehen, der in Form eines ‚Best-Practice‘-Berichts und mit Ausschnitten der Unterrichtstagebücher veröffentlicht werden könnte.

Das Konzept bietet neben den angestrebten Zielen vor allem zwei zusätzliche Möglichkeiten: 1. Das Konzept kann innerhalb der Lehramtsausbildung (z.B. innerhalb der oben erwähnten Arbeit im Schülerlabor) genutzt werden, um sowohl das Entwerfen von IniK-Einheiten, als auch Formen der strukturierten Reflexion zu üben. 2. Die strukturierte Form der Reflexion kann als Grundlage für eine Evaluation der Arbeit des Berliner IniK-Sets dienen. Vor allem die dabei entstehenden Produkte ermöglichen es, den Gestaltungsprozess zu analysieren. Im Sinne der drei vorgestellten agilen Forschungsansätze darf sich die Forschung jedoch nicht auf eine bloße empirische Überprüfung des Erfolgs der entworfenen Unterrichtseinheiten beschränken. Die Forschung sollte vielmehr darauf abzielen, den Gestaltungsprozess zu verbessern und die Erfahrungen und Erkenntnisse aus dem Unterrichtsentwicklungs-Projekt für Dritte sichtbar zu machen.

6 Literatur

- [AP07] Altrichter, H./Posch, P., Lehrerinnen und Lehrer erforschen ihren Unterricht – Unterrichtsentwicklung und Unterrichtsevaluation durch Aktionsforschung, Bad Heilbrunn 2007⁴.
- [Br92] Brown, A., Design Experiments: Theoretical and Methodological Challenges in Creating Complex Interventions in Classroom Settings, in: *The Journal Of The Learning Sciences* 2(2), S.141-178, 1992.
- [DD11] Diethelm, I./Dörge, C., Zur Diskussion von Kontexten und Phänomenen in der Informatikdidaktik, in: Thomas, Marco (Hrsg.), *Informatik in Bildung und Beruf – „Informatik und Schule – INFOS 2011“*, Bonn 2011, S.67-76.
- [DKW11] Diethelm, I./Koubek, J./Witten, H.: IniK – Informatik im Kontext, Entwicklungen, Merkmale und Perspektiven. *LOG IN* 168/170 (2011), S. 97-105.
- [EI10] Einsiedler, W., Didaktische Entwicklungsforschung als Transferförderung, in: *Zeitschrift für Erziehungswissenschaft* (13), 2010, S.59-81.
- [EP10a] Engbring, D./Pasternak, A., Einige Anmerkungen zum Begriff IniK, in: Diethelm, I./Dörge, C./Hildebrandt, C./Schulte, C., (Hrsg.): *Didaktik der Informatik*, 6. Workshop der GI-Fachgruppe >>Didaktik der Informatik<<, 2010, S. 119–124.
- [EP10b] Engbring, D./Pasternak, A., IniK – Versuch einer Begriffsbestimmung. In: Brandhofer, G./Futschek, G./Micheuz, P./Reiter, A./Schoder, K. (Hrsg.), *25 Jahre Schulinformatik in Österreich – Zukunft mit Herkunft*. Wien: Österreichische Computer Gesellschaft, 2010a, S. 100–115.
- [ER02] Eilks, I./Ralle, B., Partizipative fachdidaktische Aktionsforschung - ein Modell für eine praxisnahe curriculare Entwicklungsforschung in der Chemiedidaktik. In: *Chemie konkret* 9(1) (2002), 13-18.
- [GI08] Gesellschaft für Informatik e.V. (Hrsg.), Grundsätze und Standards für die Informatik in der Schule. *Bildungsstandards Informatik für die Sekundarstufe I*. Beilage zu *LOG IN*, 28. Jg. (2008), Heft Nr. 150/151.
- [KSW09] Koubek, J./Schulte, C./Schulze, P./Witten, H., Informatik im Kontext (IniK) – Ein integratives Unterrichtskonzept für den Informatikunterricht. In: B. Koerber (Hrsg.): *Zukunft braucht Herkunft – 25 Jahre „INFOS – Informatik und Schule“*, Bonn 2009, S. 268–279.
- [Re05] Reinmann, G., Innovation ohne Forschung? Ein Plädoyer für den Design-Based Research Ansatz in der Lehr-Lernforschung, in: *Unterrichtswissenschaft*, 33. Jg., 2005, S.52-69.
- [Sch83] Schön, D., *The Reflective Practitioner – How Professionals Think in Action*, London 1983.
- [TH98] Tulodziecki, G./Herzig, B., *Praxis- und theorieorientierte Entwicklung und Evaluation von Konzepten für pädagogisches Handeln*. Veröffentlichung der Universität Paderborn, 1998.

„Ich glaube, Google ist so was wie eine Vorhalle des Internets“ – Erste Ergebnisse einer qualitativen Untersuchung von Schülervorstellungen von der Suchmaschine Google

Oliver Seifert, Tony Sauck, Maximilian Schwarzbach,
Christopher Lerch, Martin Weinert, Maria Knobelsdorf

Fakultät für Informatik
Technische Universität Dortmund
Otto-Hahn-Str. 14
44227 Dortmund
oliver.seifert@tu-dortmund.de,
tony.sauck@tu-dortmund.de,
maximilian.schwarzbach@tu-dortmund.de,
christopher.lerch@tu-dortmund.de,
martin.weinert@tu-dortmund.de,
maria.knobelsdorf@tu-dortmund.de

Abstract: In einer qualitativen empirischen Studie wurden fünf Schüler im Alter von 13-15 Jahren zu ihren Alltagsvorstellungen von der Suchmaschine Google befragt. Die mittels Metaphern geschilderten mentalen Modelle und Erklärungsansätze der Befragten sind schlüssig und plausibel, weichen jedoch stark von den wissenschaftlichen Modellen und Konzepten ab. Damit verdeutlichen die Ergebnisse eindrucksvoll die Notwendigkeit der Entwicklung einer informatischen Bildung bei den Befragten schon zu Beginn der Sek. I. Gleichzeitig wird deutlich vor welchen Herausforderungen der Informatikunterricht steht, wenn er individuelle Vorstellungen von Schülerinnen und Schülern im konstruktivistischen Sinne berücksichtigen will.

1 Einleitung

Jugendliche interagieren heutzutage seit ihrer frühesten Kindheit mit einer Vielzahl von Informatiksystemen ohne notwendigerweise Kenntnis von deren informationstechnischen Funktionsprinzipien und den zugrundeliegenden informatischen Konzepten und Modellen zu haben. Durch die regelmäßige Interaktion mit bestimmten Systemen werden sog. *mentale Modelle* von der Funktionsweise der jeweiligen Systeme entwickelt, die die weitere Interaktion und Handlungsmöglichkeiten bestimmen, vgl. [Mo03]. Ausgehend von einem kognitiv-konstruktivistisch orientierten Verständnis von Lernprozessen, aber auch im Hinblick auf die Neuausrichtung des Unterrichts an Bildungsstandards und Kompetenzen stellen sich aus fachdidaktischer Perspektive folgende Fragen: Welche mentalen Modelle entwickeln Jugendliche von spezifischen Informatiksystemen und wie entstehen diese genau? Was passiert mit solchen

subjektiven Modellen wenn im Informatikunterricht (IU) die jeweiligen Fachkonzepte und Funktionsprinzipien vorgestellt und erarbeitet werden?

In anderen Fachdidaktiken, insbesondere den Naturwissenschaftsdidaktiken, wird den sog. Alltagsvorstellungen von Schülerinnen und Schülern (SuS) eine enorme Bedeutung für einen gelingenden Fachunterricht beigemessen, vgl. [Du10]. Das Ergebnis sind zahlreiche Studien und Forschungsprogramme, deren Ergebnisse nachhaltig zur Qualität des Fachunterrichts beitragen. In der fachdidaktischen Forschung der Informatik entsteht seit einiger Zeit ein Bewusstsein für dieses Thema. So haben Diethelm et al. (2010) SuS-Vorstellungen zum Thema Internet untersucht, vgl. [DZ10]. Rabel (2011) diskutiert den Begriff des mentalen Modells, vgl. [Ra11]. Danielsiek et al. (2012) untersuchen Fehlkonzepte von Informatik-Studierenden [DPV12]. Auch die Diskussion um kontextorientierte Ansätze, vgl. [KS09], [PV10], fokussiert auf die Lernenden und den Bezug zu ihrer Lebenswelt. In der didaktischen Rekonstruktion werden die Vorstellungen der Lernenden sogar als gleichwertig zur wissenschaftlichen Sicht gesetzt, vgl. [DD11].

An diese Diskussion anknüpfend haben wir eine qualitative empirische Studie durchgeführt und Jugendliche zu ihren mentalen Modellen von der Suchmaschine Google befragt. Dabei wollten wir untersuchen, welche mentalen Modelle bzgl. der Funktionsweise der Suchmaschine Google bei Jugendlichen vorhanden sind sowie die damit zusammenhängenden Themen der Datenspeicherung und -sicherheit. Die vorliegende Studie entstand im Rahmen des Seminars „Qualitative Forschungsmethoden im Bereich: Informatik in der Bildung“ im Bachelorstudiengang Informatik Lehramt der Technischen Universität Dortmund. Mentale Modelle von Google und Suchmaschinen wurden bisher nur im Rahmen von Usability-Studien untersucht, vgl. [KH12], [Ma08], [Zh08], nicht jedoch im Fokus fachdidaktischer Fragestellungen. Google ist die dominierende Suchmaschine im Internet, daher kann davon ausgegangen werden, dass sie SuS der Sek. I sehr vertraut ist, vgl. [MS12]. Gleichzeitig ist Informationsrecherche im Internet mittels Suchmaschinen ein wichtiges Unterrichtsthema in ITG, bzw. im IU der Sek. I an vielen Schulen in NRW. Auch wenn es für die Sek. I in NRW keine Richtlinien oder anderweitige curriculare Vorgaben für den IU gibt und folglich jede Schule ihre eigenen Lehrpläne definiert, so zielen alle Unterrichtsreihen zum obigen Thema darauf ab, den SuS ein Grundverständnis der Funktionsweise von Suchmaschinen zu vermitteln. Die von uns erfassten Vorstellungen stellen daher eine wichtige Grundlage für einen an den Lernenden und ihren subjektiven Alltagsvorstellungen ausgerichteten IU dar.

2 Theoretische Einbettung

Im Folgenden stellen wir den theoretischen Rahmen vor, in den unsere empirische Studie eingebettet ist. Wir nutzen das kognitionspsychologische Konzept des „mental Modells“ und folgen dabei weitgehend der Darstellung nach Moser (2003), vgl. [Mo03], die sich unter anderem auf Johnson-Laird (1983) bezieht, vgl. [JL83], der den Fachterminus maßgeblich geprägt hat. Zusätzlich beziehen wir uns auf Norman (2002)

und seine Darstellung mentaler Modelle im Zusammenhang mit technischen Artefakten, vgl. [No02].

Mentale Modelle stellen einen Ansatz dar, mit dem in der Kognitionspsychologie menschliches *Verstehen* konzeptualisiert wird. Dabei wird davon ausgegangen, dass Wahrnehmung und Erlebnisse als spezifische Vorstellungen mental konstruiert werden und dabei helfen Annahmen über die Konsistenz und Kongruenz zukünftiger Situationen zu bilden. Ein mentales Modell kann als eine Art innerer oder mentaler Gegenstand verstanden werden, „mit Hilfe dessen aufgrund der Analogiebeziehung zum Original (einer Verhaltens-, Funktions- oder Strukturanalogie) Aufgaben und Probleme stellvertretend und mental an diesem inneren Objekt gelöst werden können. Die exakte Struktur des mentalen Modells ist dabei nicht von Bedeutung, solange eine Analogiebeziehung zwischen Modell und Original gegeben ist.“ ([Mo03], S. 185) Dabei werden mentale Modelle jedoch nicht nur als statische Abbildung verstanden, sondern auch als mentale Simulation von Prozessen und Handlungen, wodurch sie Auswirkungen auf das weitere Handeln und Problemlöseverhalten einer Person haben können (S. 187).

In Bezug auf die Interaktion mit technischen Geräten spielen nach Norman (2002) mentale Modelle eine zentrale Rolle: Technische Geräte oder Artefakte sind üblicherweise so konstruiert, dass deren Funktionsweise durch ein Interface dem Endnutzer verborgen bleibt. In der Interaktion mit dem Interface entwickeln Endnutzer daher ein eigenes mentales Modell der Funktionsweise, indem sie ihre Handlungen am Artefakt und die dabei beobachteten Effekte zu einem kausalen Zusammenhang verknüpfen und mittels Analogienbildung auf bekannte Sachverhalte zurückführen. Das mentale Modell stellt somit einen subjektiven Erklärungsansatz der verborgenen Funktionsweise dar und ist zu unterscheiden vom sog. konzeptuellen Modell, das die fachlich korrekte Anschauung repräsentiert, vgl. ([Ra11], S. 64).

3 Aufbau und Durchführung der Studie

Mentale Modelle von Google und Suchmaschinen wurden bisher nur im Rahmen von Usability-Studien untersucht, vgl. [KH12], [Ma08], [Zh08], nicht jedoch im Fokus fachdidaktischer Fragestellungen. Dabei wurde jeweils ein qualitatives empirisches Design verwendet. Die Befragten waren Studierende oder Berufstätige. Da dies folglich die erste Studie mit Jugendlichen ist, haben wir uns entschieden, auch ein qualitatives empirisches Design zu verwenden, um das Forschungsfeld zunächst zu erschließen.

3.1 Datenerhebung

Nach Moser (2003) können mentale Modelle insbesondere durch sprachliche Äußerungen erfasst und damit der qualitativen Datenauswertung zugänglich gemacht werden, d.h. die Sprache wird als Medium verstanden, durch welches mentale Modelle konstruiert und unter Umständen überhaupt erst bewusst gemacht werden können, vgl. (S. 193). Daher haben wir als Datenerhebungsinstrument unserer empirischen Studie ein leitfadenorientiertes Interview mit mehreren Themenschwerpunkten gewählt. Das

teilstandardisierte Interview erlaubte es uns, einerseits auf unsere Forschungsfragen zu fokussieren und vergleichbares Datenmaterial zu generieren, aber andererseits auch nach Bedarf vom vorgegebenen Fragenkatalog abzuweichen und intensiver auf die geschilderten Vorstellungen einzugehen. Eine besondere Sorgfalt wurde darauf gelegt die Subjektperspektive der Jugendlichen zu erheben und keine suggestiven Fragen zu stellen, Fachbegriffe zu nennen oder Bewertungen vorzugeben.

Die Interviewlänge wurde von uns auf ca. 15-20 Minuten festgelegt. Der Interview-Leitfaden bestand aus einer Einleitung, die die Funktion hatte, die Gesprächsatmosphäre aufzubauen und den ersten Erzählimpuls bei den Befragten durch Fragen zur alltäglichen Nutzung des Internets und Erfahrungen mit der Informationsrecherche zu generieren (vgl. Anhang). Überleitend zum Kernteil des Interviews wurde nach den ersten Assoziationen mit dem Begriff Google gefragt. Im Mittelpunkt standen Fragen zur Funktionsweise der Suchmaschine Google, dem Zustandekommen der Suchergebnisse, der Datenspeicherung und ihren möglichen Gefahren. Durch Fragen bzgl. des Geschäftsmodells von Google sollten weitere Aspekte des Verständnisses der Funktionsweise der Suchmaschine erschlossen werden. Im abschließenden Teil des Interviews wurden die Jugendlichen gebeten einen 1-Seitigen Fragebogen bzgl. ihrer Vorkenntnisse und ihrer allgemeinen Vertrautheit in Bezug auf Informatik, sowie ihrem sozialen Umfeld auszufüllen.

Der Kontakt zu den befragten Jugendlichen fand zum einen über eine Realschule in Dortmund statt, mit der der Autor Sauck durch seine Nachhilfetätigkeit in Kontakt steht. Hier waren zwei Schüler der Schule bereit nach dem Unterricht an einem Interview teilzunehmen. Durch Nachhilfetätigkeit des Autors Seifert konnten drei weitere Jugendliche, die ein Gymnasium in Dortmund besuchen, für das Interview gewonnen werden. Die Interviews wurden von den Autoren Sauck und Seifert durchgeführt und akustisch aufgenommen. Zum Zeitpunkt des Interviews sind die interviewten fünf Schüler 13 bis 15 Jahre alt. Ihre tägliche Internetnutzung ist geprägt durch Unterhaltungsmedien wie Musik, Videos und Online Spiele, sowie kommunikative Tätigkeiten, wie Internet-Telefonie oder soziale Netzwerke. Daneben nutzen die Jugendlichen ihren Computer und das Internet für schulische Arbeiten, Informationsrecherche und zum Programmieren. Zudem wachsen die Jugendlichen alle in einem IT-affinen Umfeld auf, bei dem mindestens ein Freund oder Familienmitglied im Bereich der Informatik arbeitet oder studiert. Eine informationstechnische Ausbildung oder entsprechendes Studium wird von den Befragten als eine mögliche Berufswahl in Betracht gezogen.

3.2 Datenauswertung

Die erhobenen Interviews wurden transkribiert und anschließend kodiert, dabei orientierten wir uns an der Technik des offenen und axialen Kodierens der Grounded Theory nach Strauß und Corbin (1996), vgl. [St96]. Hierbei haben wir zunächst in Teilgruppen die Interviews einzeln offen kodiert und erste Codes sowie beobachtbare Konzepte entwickelt. Im zweiten Schritt wurden diese Ergebnisse in der gesamten Gruppe zusammengetragen und iterativ weiter zu Kategorien verdichtet. Abschließend

fand eine Interpretation der Ergebnisse statt. Diese Methode ermöglichte es uns, aus dem Forschungsfeld heraus die mentalen Modelle aus der Subjektperspektive der Jugendlichen zu rekonstruieren, vgl. ([St96], S. 64 ff). Ein solcher Dateninterpretationsprozess kann durch das Hinzuziehen von bestehenden Theorieansätzen begleitet werden. Bei der Beschreibung ihrer eigenen mentalen Modelle nutzen Personen häufig sprachliche Metaphern oder metaphorische Ausdrücke, da diese als sprachliches Instrument eine Analogiebildung erlauben, vgl. ([Mo03], S. 188ff). Die von uns interviewten Schüler nutzten tatsächlich sehr häufig Metaphern, um ihre Vorstellungen und Verständnis von Google und dessen Funktionsweise zu erklären. Daher haben wir uns auch in der Datenauswertung auf Metaphern bzw. metaphorische Ausdrücke fokussiert.

4 Ergebnisse der Studie

Im Folgenden stellen wir vor, wie sich die befragten fünf Schüler die Funktionsweise der Suchmaschine Google erklären, welche mentalen Modelle sie diesbezüglich beschreiben und welche Metaphern sie dabei verwenden. Die Ergebnisse orientieren sich thematisch dabei an unserem Interviewleitfaden, d.h. neben der eigentlichen Funktionsweise der Suche werden auch Datenhaltung und Datenschutz thematisiert.

Vorab lässt sich zunächst festhalten, dass alle fünf Befragten die Vorstellung haben, dass Google entweder das ganze Internet darstellt oder zumindest dessen Struktur in massiver Form mitbestimmt. Ein befragter Schüler verdeutlicht dies folgendermaßen: *„Also ich denke wenn man [...] das Internet öffnet, kommt man ja eigentlich sofort zu Google. Und ich schätze dass man dann im Prinzip [...] dass Google [...] sowas wie [...] eine Vorhalle ist.“* (TA, S. 6, Z. 28). Auch die Anderen nennen Google als den ersten Anlaufpunkt im Internet, nicht nur wenn es um das Suchen von Informationen geht. Das Ausweichen auf andere Suchdienste kommt für sie nicht in Frage, da sie die Qualität der durch andere Suchmaschinen generierten Suchergebnisse anzweifeln. Begründet wird das durch schlechte Suchergebnisse oder durch die ungewollte Installation von Toolbars durch die Suchmaschine Yahoo.

4.1 Funktionsweise der Suche

Bei einer Suchanfrage gehen alle Befragten zunächst davon aus, dass Google Webseiten vorab auf eine bestimmte Art und Weise bewertet. Zur tatsächlichen Bewertungsvorgehensweise bestehen folgende Vorstellungen:

1. Die Häufigkeit, mit der eine Webseite insgesamt bisher besucht wurde, bestimmt die Bewertung.
2. An Google wird eine Gebühr pro Webseite entrichtet und deren Höhe entscheidet über die Platzierung des Webseitenlinks in der Ergebnisliste.
3. Benutzer bewerten einzelne Webseiten und entscheiden so über ihre Wertigkeit.

Keiner der Befragten nennt hier die Bewertung mittels des von Google tatsächlich verwendeten PageRank-Algorithmus [BC06], der auch die Grundlage für Googles Geschäftsmodell darstellt. Die Befragten erklären, dass Google an erster Stelle Einnahmen durch Werbung generiert, wie es bspw. durch vorangestellte Werbung bei Videos auf YouTube oder durch Anzeigen am Rand der Suchergebnisse geschieht. Die oben genannte Methode, die Reihenfolge der Ergebnisse durch Gebühren zu bestimmen, sei ebenfalls eine Möglichkeit wie Google Gewinne erzielt. Zwei Schüler erklären, dass Google für jeden Klick auf einen dargestellten Webseitenlink pauschal einen Cent verdient: *„Ja, ich hab das im Fernsehen gesehen. Und die meinten, durch einen Klick kriegen die immer einen Cent.“* (FK, S. 4, Z. 21).

Die eigentliche Suche wird mit unterschiedlichen Modellen erklärt. Ein Befragter geht davon aus, dass Webseiten (nach ihrer Bewertung durch eine der oben beschriebenen Bewertungsmethoden) von Google in einer Art Rangliste geführt werden, wobei zu jedem möglichen Suchbegriff eine eigene Liste von Webseiten zugeordnet ist. Dies wird mit der Metapher „Schlüssel-Schloss-Prinzip“ beschrieben. Der Suchbegriff ist bei der Suchanfrage eine Art „Schlüssel“, während die Suchbegriffslisten „Fächer“ darstellen. Bei der Suchanfrage wird zum Schlüssel das passende Fach gesucht: *„[...] wie z.B. so Schlüssel-Schloss-Prinzip, ganz viele kleine Fächer mit Schlössern dran, wenn man irgendwas eingibt sozusagen als Schlüssel, dass das dann nur in ein Ding passt, wo halt die einzelnen Webseiten dann sind.“* (TA, S. 4, Z. 5).

Ein anderer Befragter greift ebenfalls auf die Rangliste zurück und erklärt, dass frühere Suchanfragen gespeichert und dann wiederverwendet werden: *„[...] es kann natürlich auch sein, dass Google sich merkt, was andere Nutzer eingegeben haben und greift das auch bei anderen Suchen auf“* (LS, S. 4, Z. 25ff.). Ein Befragter beschreibt seine Vorstellung von Google durch Analogien zu einer Telefonzentrale nach dem Client-Server-Modell. Hier werden Anfragen serverseitig entgegen genommen, autonom überprüft, ausgewertet und die relevantesten Ergebnisse aufbereitet dargestellt: *„[...] das geht in so eine Zentrale, dann wird das überprüft [...], dann wird das gezeigt [...].“* (FK, S. 2, Z. 6).

Die Befragten haben auch relativ konkrete Vorstellungen darüber, wie die eigentliche Suchanfrage von Google umgesetzt wird. So erklärt ein Befragter, dass die Suche nicht ausschließlich auf Google eigenen Rechnern, sondern auch auf dem Rechner des Nutzers abläuft: *„[...] wenn du Google installierst, also wenn du da was eingibst, dann installiert Google deinen Rechner und dann wird das immer dahin geschickt dann.“* (FK, S. 4, Z. 1f.). Daher würde auch die Geschwindigkeit der Webseitensuche entsprechend von der benutzereigenen Hard- und Software abhängen.

4.2 Datenspeicherung

Die Befragten haben alle eine grobe Vorstellung davon was Daten sind, die sie als Zahlen beschreiben. Es besteht auch die Vorstellung, dass Daten einen größeren Umfang haben und irgendwo physikalisch gespeichert werden. Es fällt auf, dass die Befragten um die erhebliche Datenflut des Internets Bescheid wissen: *„Wobei das Internet ist ja eigentlich riesig, das passt doch nie im Leben auf so viele Festplatten.“* (TA, S. 7, Z. 18). Ein

Befragter geht noch einen Schritt weiter und nennt neben Servern, das vermeintliche Speichermedium, auch Datenbanken (wobei eine weitergehende Erklärung nicht erfolgt). Die angesprochenen Server stehen dabei, laut Aussage des Befragten, in großen Serverräumen und bezieht sich auf die im vorhergehenden Unterabschnitt beschriebene Metapher der Zentrale: „[...] also ich könnte mir vorstellen, dass diese ganzen Anlagen, diese ganzen Speicher, ja, Festplatten eben, auf dieser großen Anlage alle miteinander vernetzt sind“ (TA, S. 6, Z. 4-5).

Es ist den Schülern bekannt, dass Server Abwärme produzieren und dies scheint für sie relevant zu sein, da sie dieses Thema selbstständig ansprechen und die Standortwahl der Server weiter diskutieren: „[...] Das sind dann wahrscheinlich so große Serverräume, mit mehreren Kühltürmen, weil die erzeugen ja eine enorm große Wärme. Dann muss das alles so gebündelt werden und dann ordentlich gekühlt werden... weil sonst überhitzt ja alles.“ (LS, S. 8, Z. 27-30). Als Lösungsmöglichkeit sieht ein Befragter außerdem die Wahl des Serverstandortes in kalten Regionen: „Irgendwo vielleicht in der Antarktis, na gut Antarktis nicht unbedingt, aber da oben irgendwo am Nordpol weils da kalt ist. Oder Norwegen.“ (MC, S. 7, Z. 19-20).

4.3 Datenschutz

Das Thema Datenschutz ist den befragten Schülern bekannt. Sie erklären, dass Google „über alles und jeden“ Daten sammelt, das Verhalten der Benutzer mithilfe der Suchfunktion im Internet mit verfolgt und jederzeit weiß, welche Person welche Website besucht. Dies wird entsprechend mit dem eigenen mentalen Modell bzgl. der Funktionsweise plausibilisiert: „Sie wissen halt, dass ich auf diese Website gegangen bin. [...] ich glaub schon, dass die das dann noch wissen dass ich dann noch am PC bin. Also nicht, wenn ich es geschlossen habe, sondern wenn ich auf dieser Website bin. Weil, das ist ja auch im Prinzip eine von Ihren Kammern dann.“ (TA, S. 12, Z. 6). Wie in den vorhergehenden Unterabschnitten beschrieben wurde, stellen sich die Befragten vor, dass Google Webseiten physikalisch auf großen Serveranlagen speichert. Bei einer Suchanfrage wird auf diesen Datenbestand zugegriffen, um weitere personenbezogene Informationen zu beziehen.

Die befragten Schüler bewerten Googles Kenntnis über das eigene Such- und Surfverhalten dennoch insgesamt als positiv: „Ja, das ist eigentlich gut, weil... wenn man dann Google benutzt, hat man dann natürlich [...] ich sag mal „besten“ Ergebnisse, also.“ (LS, S. 12, Z. 11). Gerade dieser Schüler betont auch sein Vertrauen in Google und dass die Datensammlung den Suchergebnissen zugutekommt. Ein anderer befragter Schüler sieht darin aber auch einen Eingriff in die Privatsphäre. Möglichkeiten zum Schutz der eigenen Daten sind den Schülern bekannt, sei es in Form eines sog. Inkognito Modus des Browsers oder mittels Löschen erzeugter Daten.

Nicht nur der Schutz der eigenen bei der Suche erzeugten Daten vor Google, sondern auch der Schutz der Webseitenbetreiber wurde von den Schülern im Interview thematisiert. Ein Schüler erklärt, dass Webseitenbetreiber ihre Inhalte vor Suchmaschinen schützen können, indem sie ihre Webseite besonders kennzeichnen. Hier zeigt sich eine weitere Metapher, welche Analogien zur Markenzeichnung und

Schutz des Urheberrechts aufweist. Dies würde, laut befragtem Schüler, Google signalisieren, dass es die Seite nicht anzeigen kann: „[...] dass man sich halt schon so ein Schutzprogramm auf seine selbsterstellte Website zieht, dass halt nur Google zum Beispiel mit Genehmigung drauf kommt. [...] Bei Büchern oder bei Namen ist das ja so, da kann man einfach „tm“ hinten dran machen und äh vielleicht, dass es sowas ähnliches bei einer Website dann auch gibt, wenn der Hersteller das nicht will.“ (TA, S. 9, Z. 28-29; S. 10, Z. 2-4). Als weitere Alternative wird die Möglichkeit genannt, Webseiten und ihre Inhalte durch Passwörter zu schützen. Dennoch vermutet der befragte Schüler, dass Google die Kapazitäten besitzt, um jedes beliebige Passwort zeitnah entschlüsseln zu können. Das Unternehmen würde jedoch diesen Schritt nicht wagen, da es sonst eine Rufschädigung in der öffentlichen Wahrnehmung zu Folge hätte.

5 Diskussion der Ergebnisse

In diesem Abschnitt diskutieren wir die vorgestellten Ergebnisse unserer Studie und leiten daraus mögliche Praxisempfehlungen für den IU ab.

Uns fiel auf, dass die befragten Schüler ihre mentalen Modelle erst im Interview zu entwickeln schienen und mit jeder weiteren Frage nach und nach ergänzten. Es schien, als ob die Schüler zuvor nie mit dem Thema konfrontiert worden wären. Gleichzeitig entstanden dennoch unterschiedliche und aussagekräftige Vorstellungen. Zusätzlich bestätigen alle Befragten ihre intensive und jahrelange Internet- und Computernutzung. Im Hinblick auf unsere theoretischen Grundlagen, gehen wir daher davon aus, dass gewisse mentale Modelle implizit bei den Befragten durch ihre Interaktion mit dem System angelegt waren, aber erst durch unser Interview eine bewusste Auseinandersetzung damit angeregt wurde.

Weiterhin fiel uns auf, dass die Erklärungsansätze und damit verbundenen mentalen Modelle der befragten Schüler weitgehend schlüssig und plausibel sind, obwohl sie gleichzeitig stark vom jeweiligen wissenschaftlichen, also konzeptuellen Modell abweichen. Bei Betrachtung der von den Schülern verwendeten Begriffe fällt auf, dass viele Fachtermini angesprochen werden, obwohl die konzeptuellen Modelle den Schülern nicht bekannt zu sein scheinen. Begriffe wie „Server“, „host“ oder „Service“ werden zwar von den Jugendlichen verwendet, aber ein tatsächliches Verständnis dieser fehlt. Weiterhin schien es den befragten Schülern schwer zu fallen zwischen der Suchmaschine Google und dem Internet zu unterscheiden. Ein Schüler beschrieb Google als Vorhalle des Internets, für einen anderen war das Internet die Summe aller Suchmaschinen. Auffällig war, dass die Interviewten Googles Datensammlung als potentiell ungefährlich einstufen. Sie wussten zwar, dass Daten von Google gesammelt werden und dass diese personenbezogen sein können, glaubten insgesamt jedoch nicht, dass sie selbst oder ihre Daten dadurch gefährdet sein könnten. Insgesamt bestätigen diese Ergebnisse damit eindrucksvoll die Notwendigkeit informatischer Bildung bei den Jugendlichen. In einem konstruktivistisch begriffenen und an Standards und Kompetenzen orientierten IU müssten zum Thema Suchmaschinen entsprechend die individuellen Vorstellungen der SuS aufgegriffen und durch entsprechende Reflexions-

und Erarbeitungsphasen weiter entwickelt werden, hin zu einem informatischen Grundverständnis.

Vor dem Hintergrund der vielfältigen und vom konzeptuellen Modell teilweise stark abweichenden Vorstellungen der Befragten, legen unsere Ergebnisse nahe, dass eine Bewusstmachung der eigenen Vorstellungen wichtig für die Lernenden ist und als Aktivierungsphase entscheidend zum Lernerfolg beitragen könnte. So wäre es möglich zu Beginn einer Unterrichtsreihe die SuS in Gruppenarbeit erarbeiten zu lassen, welche Vorstellungen sie bzgl. Suchmaschinen haben. Die dabei konstruierten Modelle könnten zusammengetragen und in der Klasse gemeinsam besprochen werden. Die Lehrperson könnte dann im weiteren Verlauf des Fachunterrichts immer wieder auf die individuellen Modelle zurückgreifen und versuchen Brücken zwischen den Schülervorstellungen und den Fachkonzepten herzustellen.

Wir vermuten, dass das Hintergrundwissen der befragten Schüler in hohem Maße von den Medien beeinflusst wird. Insbesondere dann, wenn die Schüler selber keine eigene oder bessere Erklärung hatten, beriefen sie sich auf Informationen, wie sie durch öffentliche Medien kommuniziert werden. Die Lehrperson könnte diesen Umstand in ihrem Unterricht nutzen, in dem sie auch solche Aspekte in den Unterricht mit aufnimmt und sie dann kritisch diskutiert. Dies kann durchaus als Ankerpunkt verwendet werden, um ein neues Themengebiet zu erschließen oder zu einem alten zurückzukehren, nachdem die SuS ihr Verständnis soweit hinterfragt haben, dass gewisse Widersprüchlichkeiten aufgedeckt werden können. Damit würde gleichzeitig sicher gestellt, dass sich die SuS mit ihrem eigenen mentalen Modell weiter auseinandersetzen und dieses kritisch reflektieren je mehr Fachwissen sie dazulernen. Uns ist bewusst, dass dies eine große Herausforderung für die Lehrperson darstellt, da Individuen ihre subjektiven Alltagsvorstellungen nicht automatisch zu Gunsten des Fachmodells aufgeben, vgl. [DT03]. Daher verstehen wir unsere Vorschläge für den IU als Anregung sowie Ausblick auf weitere fachdidaktische Forschungstätigkeiten auf diesem Gebiet.

6 Fazit

In diesem Artikel haben wir die Ergebnisse einer qualitativen empirischen Studie vorgestellt, bei der wir fünf Schüler im Alter von 13-15 Jahren zu ihren Alltagsvorstellungen von der Suchmaschine Google durch ein Leitfadeninterview befragt haben. Die mittels Metaphern geschilderten, mentalen Modelle und Erklärungsansätze der Befragten sind schlüssig und plausibel, weichen jedoch stark von den wissenschaftlichen Modellen und Konzepten ab. Unsere Ergebnisse verdeutlichen eindrucksvoll die Notwendigkeit einer informatischen Bildung bei den Befragten. Vor dem Hintergrund der Allgegenwärtigkeit von Informatiksystemen, bzw. die Nutzung dominierender Anwendungen wie Google, ist ein reguläres Schulfach Informatik schon zu Beginn der Sek. I mehr als notwendig. Gleichzeitig wird deutlich vor welchen Herausforderungen Lehrende im Informatikunterricht stehen, wenn sie die individuellen Vorstellungen und Modelle der Schülerinnen und Schüler konstruktivistisch in den Unterricht einbeziehen wollen.

7 Danksagung

Wir bedanken uns bei allen befragten Schülern unserer Studie, sowie ihren Eltern und dem Schulleiter der Realschule für ihre Unterstützung.

8 Literatur

- [BC06] Benincasa, C., Calden, A., Hanlon, E., Kindzerske, M., Law, K., Lam, E., Rhoades, J., Roy, I., Satz, M., Valentine, E. und Whitaker, N.: Page Rank Algorithm. Department of Mathematics and Statics, University of Massachusetts, Amherst, Research, 2006.
- [DPV12] Danielsiek, H., Paul, W. und Vahrenhold, J.: Detecting and understanding students' misconceptions related to algorithms and data structures. Proceedings of the 43rd ACM technical symposium on Computer Science Education (SIGCSE), ACM, S. 21-26, 2012.
- [DT03] Duit, R. und Treagust, D. F.: Conceptual change: A powerful framework for improving science teaching and learning, International Journal of Science Education, 25, 6, 2003.
- [Du10] Duit, R.: Schülervorstellungen und Lernen von Physik. Piko-Briefe – der fachdidaktische Forschungsstand kurzgefasst, IPN Kiel, 2010.
- [DZ10] Diethelm, I. und Zumbärgel, S.: Wie funktioniert eigentlich das Internet? - Empirische Untersuchung von Schülervorstellungen. Tagungsband des 6. GI-Workshops der Fachgruppe Didaktik der Informatik, Köllen Verlag, S. 233-44, 2010.
- [DD11] Diethelm, I., Dörge, C., Mesaros, A. und Dünnebier, M.: Die Didaktische Rekonstruktion für den Informatikunterricht. Tagungsband der 14. Fachtagung Informatik und Schule (INFOS), Köllen Verlag, S. 77-86, 2011.
- [JL83] Johnson-Laird, P.: Mental Models. Harvard University Press, 1983.
- [KH12] Khoo, M. und Hall, C.: What Would 'Google' Do? Users' Mental Models of a Digital Library Search Engine. Lecture Notes in Computer Science, 2012.
- [KS09] Koubek, J., Schulte, C., Schulze, P. und Witten, H.: Informatik im Kontext (IniK) - Ein integratives Unterrichtskonzept für den Informatikunterricht. Tagungsband der 13. GI-Fachtagung Informatik und Schule (INFOS), Köllen Verlag, S. 268-279, 2009.
- [Ma08] Martzoukou, K.: Students' Attitudes Towards Web Search Engines – Increasing Appreciation of Sophisticated Search Strategies. Libri, Vol. 58, S. 182-201, 2008.
- [Mo03] Moser, K. S.: Mentale Modelle und ihre Bedeutung: kognitionspsychologische Grundlagen des (Miss-)Verstehens. Schriften zur Symbolforschung, Vol. 13, Bern: Peter Lang, S. 181-205, 2003.
- [MS12] Medienpädagogischer Forschungsverbund Südwest: JIM Studie 2012: Basisuntersuchung zum Medienumgang 12- bis 19-Jähriger in Deutschland, Stuttgart, 2012.
- [No02] Norman, D.: The Design of Everyday Things, Perseus Books, 2002.
- [PV10] Pasternak, A. und Vahrenhold, J.: Braided Teaching in Secondary CS Education: Contexts, Continuity, and the Role of Programming. Proceedings of the 41st ACM SIGCSE conference, ACM, S. 204-208, 2009.
- [Ra11] Rabel, M.: Grundvorstellungen in der Informatik. Praxisbeiträge zur 14. Fachtagung Informatik und Schule (INFOS), Köllen Verlag, S. 61-70, 2011.
- [St96] Strauss, A. und Corbin, J.: Grounded Theory – Grundlagen Qualitative Sozialforschung. Psychologie Verlags Union, Weinheim, 1996.
- [Zh08] Zhang, Y.: Undergraduate students' mental models of the Web as an information retrieval system. J. Am. Soc. Inf. Sci. Technol. 59, 13, 2008.

9 Anhang Interviewleitfaden und Fragebogen

Einleitende Worte:

Wir führen heute eine Studie im Rahmen eines Informatik Seminars an der TU Dortmund durch. Bei der Studie interessieren wir uns für die Vorstellung von Jugendlichen von Medien und dem Internet in ihrem Alltag. Lass dir so viel Zeit wie du brauchst, um die Fragen zu beantworten. Es geht um deine Vorstellungen, nicht darum was richtig oder falsch ist.

Zu der Durchführung: Wir werden gleich unser Gespräch mit dem Laptop aufzeichnen und im Anschluss anonym verschriftlichen. Das Interview wird also nicht in Verbindung mit deiner Person gebracht werden können. Weiterhin wird natürlich alles streng vertraulich behandelt. Deine Eltern haben das Infoblatt gelesen und unterschrieben.

1. Einleitung

„Erzähl doch mal, was machst du eigentlich so im/mit dem Internet?“
„Was assoziiert du mit Google?“ bzw. „Was macht Google alles?“

2. Fokus auf Suchfunktion

[Stichwort: Allgemein Suchmaschine]

„Wie funktioniert Googles Suchmaschine eigentlich?“
„Was passiert zwischen dem Drücken der Tasten und dem Anzeige der Ergebnisse?“

[Stichwort: Suche]

„Wie/Woher werden Daten im Internet gesammelt?“
(Wenn Antwort Datenbanken:) „Wie stellst du dir diese Datenbank vor?“
„Wie kommen die Daten in Googles Datenbank?“

[Stichwort: Auswertung, Algorithmus]

„Warum ist Wikipedia meistens ganz oben in den Ergebnissen?“
„Zeigt Google jedem die gleichen Ergebnisse?“
(je nach Antwort) „Wie erklärst du dir die Unterschiede?“, „Wie bewertest du das?“

4. Speicherung von Daten

„Wo speichert Google die ganzen Daten?“
„Wie werden die Daten gespeichert?“
„Wie lange werden die Daten gespeichert?“
„Was könnte Google noch mit den Daten anstellen außer für die Suchfunktion?“
„Wie könntest du verhindern, dass Google deine Daten sammelt?“

5. Google als Unternehmen

„Womit verdient Google eigentlich sein Geld?“

Schriftlicher Teil zur Befragung

Alter in Jahren:					
Geschlecht:					
<input type="checkbox"/>	Männlich	<input type="checkbox"/>	Weiblich		
Erfahrungen im Umgang mit dem Internet und Computern					
Das Internet nutze ich zum :					
	Trifft gar nicht zu	Trifft kaum zu	Trifft zu	Trifft voll zu	
Arbeiten für Schule (Recherche, Textverarbeitung ...)					
Medienkonsum (Videos, Bilder, eBooks)					
Soziale Netzwerke und Kommunikation (Facebook, Skype ...)					
Computerspiele					
Etwas anderes: _____					
Wie viele Stunden verbringst du täglich im Internet?					
<input type="checkbox"/>	1 - 3	<input type="checkbox"/>	4 - 5	<input type="checkbox"/>	Mehr als 5
Hattest/Hast du Informatik in der Schule?					
<input type="checkbox"/>	Ja	<input type="checkbox"/>	Nein		
Falls ja, welcher Kurs bzw. in welcher Schulstufe?					
Kannst du Programmieren?					
<input type="checkbox"/>	Ja	<input type="checkbox"/>	Nein		
Falls ja, wo hast du Programmieren gelernt? (Mehrfachankreuzen möglich)					
<input type="checkbox"/>	In der Schule	<input type="checkbox"/>	Hobbymäßig		
Falls ja, in welchen Sprachen?					
Soziales Umfeld					
In meiner Familie arbeitet jemand in der IT/Informatikbranche:					
<input type="checkbox"/>	Ja	<input type="checkbox"/>	Nein		
Falls ja, wer und was macht dieses Familienmitglied genau?					
In meinem Freundeskreis ist mindestens ein „IT-Spezialist“, der/die auf viele computerbezogene Fragen einen Rat weiß:					
<input type="checkbox"/>	Ja	<input type="checkbox"/>	Nein		
Falls ja, wer und was macht dieser Freund?					
Weißt du schon, welchen Beruf/Ausbildung du nach der Schule/nach dem Abitur machen möchtest?					

Grundvorstellungen zu Klassen und Objekten

Magnus Rabel

Institut für Didaktik der Mathematik und Informatik
Johann Wolfgang Goethe-Universität Frankfurt am Main
Senckenberganlage 9
60325 Frankfurt am Main
rabel@math.uni-frankfurt.de

Abstract: Im Rahmen eines Schulversuchs zu genetischem Informatikunterricht wurden von den beteiligten Schülerinnen und Schüler Grundvorstellungen zu Konzepten der Objektorientierten Programmierung erhoben. Diese Erhebung gliederte sich in zwei Phasen. Direkt nach der Unterrichtseinheit im Schuljahr 2011/2012 wurden die Schülerinnen und Schüler gebeten, Begriffe der Objektorientierung schriftlich zu beschreiben. Nach neun Monaten, in denen objektorientierte Konzepte im Unterricht keine zentrale Rolle spielten, wurde in einer zweiten Befragung nach dem gleichen Design erhoben, welche der zuvor untersuchten Begriffe in welcher Form verinnerlicht worden waren. Eine qualitative Analyse der Schülerarbeiten ergab eine vielversprechende Anzahl von Schülervorstellungen sowie interessante Folgerungen zur Entwicklung von Grundvorstellungen, die weiterer Untersuchungen bedürfen.

1 Motivation

Das Lernen objektorientierter Konzepte ist nach wie vor ein aktuelles Thema im didaktischen Diskurs sowie im schulischen Alltag. Viele Einflüsse bestimmen den Erfolg der Lernenden, wie bspw. in [ES09] dargestellt. Ein Aspekt ist (nicht nur im Zusammenhang mit Objektorientierung) die Ausbildung geeigneter mentaler Modelle (vgl. [No83], zitiert in [Ra11]), da erfolgreiches Lernen stark davon abhängt, dass Lernende geeignete, also tragfähige und langlebige Modellvorstellungen von Lerninhalten entwickeln (vgl. bspw. [CLB07]), deren Untersuchung sich jedoch als schwierig erweist (vgl. bspw. [Du94], [Ma07]). Einen Lösungsansatz bietet das Konzept der informatischen Grundvorstellung, das in [Ra11] vorgestellt worden ist und im nächsten Abschnitt kurz umrissen wird.

Im Informatikunterricht der Oberstufe stoßen Lehrkräfte auf verschiedenste Herausforderungen, da zum einen die im Lehrplan vorgesehene Zeit für die Einführung in die Objektorientierte Modellierung relativ knapp bemessen ist und zum anderen Projekte schwierig zu finden sind, deren Komplexität die Umsetzung in einem objektorientierten Paradigma rechtfertigen (vgl. [Ko09]).

In Anbetracht der genannten Aspekte wurde für uns die Frage interessant, ob und wie sich Grundvorstellungen erheben und klassifizieren lassen und welche Vorstellungen Informatikschülerinnen und -schüler unter den unterrichtlichen Rahmenbedingungen entwickeln, wie tragfähig diese sind, und wie gut sie auch nach längerer Zeit noch abrufbar sind. Die Abrufbarkeit hat mehrere Dimensionen. Hier wird nur auf die Fähigkeit, die Konzepte Klasse und Objekt zu beschreiben, eingegangen. Wie gut die Schülerinnen und Schüler auch nach längerer Zeit mithilfe dieser Konzepte Problemlösungen modellieren können, wird Thema einer anderen Untersuchung sein (vgl. Abschnitt 5).

2 Grundvorstellungen in der Informatik

Wie bereits erwähnt gilt die Untersuchung mentaler Modelle als äußerst schwierig. Das Konzept der informatischen Grundvorstellung (vgl. [Ra11]) ist unseres Erachtens sehr gut geeignet, mentale Modelle von Schülerinnen und Schülern ausreichend genau zu beschreiben und darüber hinaus das Lernen komplexer informatischer Inhalte zu strukturieren. Für diesen Artikel werden hier die charakterisierenden Aufgaben und Leistungen von Grundvorstellungen informatischer Inhalte genannt, die auf den Charakterisierungen mathematischer Inhalte nach Rudolf vom Hofe (vgl. [Ho95]) basieren:

- Die Sinnkonstituierung eines Begriffs durch Anknüpfung an bekannte Sach- oder Handlungszusammenhänge bzw. Handlungsvorstellungen,
- der Aufbau entsprechender (visueller) Repräsentationen bzw. "Verinnerlichungen",
- die Fähigkeit zur Anwendung eines Begriffs auf die Wirklichkeit durch Erkennen der entsprechenden Struktur in Sachzusammenhängen.

Wir gehen, wie im Folgenden deutlich wird, davon aus, dass Grundvorstellungen verschiedene Perspektiven auf das gleiche Konzept bzw. das gleiche mentale Modell eröffnen. Mehrere Grundvorstellungen beschreiben also zusammengenommen die wichtigsten charakteristischen Aspekte eines mentalen Modells.

3 Organisatorischer Rahmen und Forschungsdesign

Die Untersuchungen fanden in zwei Kursen an zwei verschiedenen hessischen Gymnasien statt, die im Rahmen eines Schulversuchs zu genetischem Informatikunterricht der gymnasialen Oberstufe unterrichtet wurden. Dabei handelte es sich um dreistündige Informatikgrundkurse mit 5 Schülerinnen und 7 Schülern (Kurs 1) bzw. 14 Schülern (Kurs 2) im ersten reinen G8-Jahrgang der betreffenden Schulen. Es wurden in beiden Kursen zwei Befragungen mit Freitextfragebögen durchgeführt. Die erste Befragung fand gegen Ende der Einheit zur Objektorientierten Modellierung im März/April 2012 und die zweite Befragung im Dezember 2012 statt.

Dem Unterricht lag der genetische Ansatz nach Erich Wittmann (vgl. [Wi81]) zugrunde, der von uns auf den Informatikunterricht angewendet wurde. Daraus resultierten eine moderate Anpassung des Lehrplans und die Wahl einer anderen Programmiersprache. Python erwies sich als die geeignete Wahl, da der Einstieg in die Programmierung und die Steigerung der Komplexität der Projekte sehr intuitiv gelingen konnte (vgl. dazu [Sc11] und [ORS12]).

Die Notwendigkeit, Objekte und Klassen einzuführen, entwickelte sich aus Problemstellungen, die zunächst mit der Datenstruktur Liste unbefriedigend bzw. unkomfortabel gelöst wurden. Den Einstieg bildete in beiden Kursen die Realisierung einer rudimentären z-Buffer-Lösung. Es sollten geometrische „Pappfiguren“ modelliert werden, die man übereinander zeichnen konnte. Das bedeutete also, dass zu den üblichen Attributen wie geometrische Maße, Farbe, Füllung etc. noch die Position in einem „Pappfigurenstapel“ hinzukam. Die Modellierung mit Listen war in vielerlei Hinsicht unkomfortabel: Man musste sich merken, an welcher Stelle in der Liste welches Attribut definiert war, benötigte eine Fallunterscheidung bei einer externen Zeichenmethode etc..

Die objektorientierte Umsetzung bot dann also die Verwendung sinnvoller Attributnamen und die Implementierung objektbezogener Zeichenmethoden, die eine Fallunterscheidung überflüssig machte. Anschließend wurde in beiden Kursen ein Bruchrechner und die Klasse Binärbaum für den TreeSort-Algorithmus modelliert und implementiert. Ab diesem Zeitpunkt unterschied sich das weitere Vorgehen in den beiden Kursen:

In Kurs 1 wurde ein Rechnernetz mit Objekten vom Typ Rechner, Switch, Netzwerkkarte etc. modelliert, teilweise implementiert und analysiert. Darauf folgten die erste Befragung und anschließend die Klausur. Im Anschluss an die Klausur wurde die bestehende Implementierung des Rechnernetzprojektes verfeinert und erweitert.

In Kurs 2 blieb die objektorientierte Modellierung des Rechnernetzes aus und eine etwas veränderte Klausur wurde vor der ersten Befragung geschrieben. Wie sich dieser Unterschied im Vorgehen in beiden Kursen eventuell auf die Ergebnisse der Befragung ausgewirkt hat, wird im nächsten Abschnitt thematisiert.

Insgesamt wurden während der gesamten Unterrichtseinheit zu OOM und OOP zentrale Begriffe wie Klasse, Objekt etc. bewusst en passant eingeführt. D.h. nur durch den Gebrauch im Unterricht lernten die Schülerinnen und Schüler die Begriffe zu identifizieren und zu verwenden. Auf eine auswendig zu lernende Definition wurde verzichtet. Dies ermöglichte eine Abfrage von Beschreibungen, die arm an nicht verstandenen auswendig gelernten Merksätzen war. Stattdessen mussten die Schülerinnen und Schüler eigene Formulierungen finden, die eine Interpretation von deren tatsächlichen Vorstellungen möglich machen sollten. Eine genauere Definition der Begriffe fand dann im Anschluss an die erste Befragung statt. Nach dem Ende der Unterrichtseinheit wurden Konzepte der Objektorientierung höchstens noch am Rande (bspw. bei der Einführung der Entity-Relationship-Modellierung) thematisiert.

Wie bspw. in [Ma07] beschrieben, ist die Beschreibung von Konzepten ein möglicher Weg, Schülervorstellungen zu erheben. So wurde für beide Befragungen in beiden Kursen wurde ein Fragebogen eingesetzt, der die folgenden Aufträge enthielt: „Beschreiben Sie prägnant, was eine **Klasse** ist, was im Gegensatz dazu ein **Objekt** ist, was **Attribute** sind, was **Methoden** sind, was ein **Konstruktor** ist.“ Anschließend an jede Teilfrage stand den Schülerinnen und Schülern ein Textfeld zur Verfügung, das sie für ihre Beschreibungen nutzen konnten. Für den gesamten Fragebogen standen 20-30 Minuten zu Beginn bzw. in der Mitte einer Unterrichtsstunde zur Verfügung, sodass eine vorzeitige Abgabe des Fragebogens nicht dazu führen konnte, dass einzelne Schülerinnen oder Schüler früher den Unterricht verlassen durften. Die Erwartung ausführlicher und aussagekräftiger Beschreibungen wurde zumindest teilweise erfüllt.

In diesem Artikel wird die Untersuchung der Schülervorstellungen bzgl. des Zusammenhangs der Konzepte Klasse und Objekt eingeschränkt. Für die Auswertung der Beschreibungen wurden drei mögliche Schülervorstellungen postuliert:

- Die Mengenvorstellung: Eine Klasse fasst strukturgleiche Objekte zusammen, hat also eine ordnende Funktion.
- Die Bauplanvorstellung: Eine Klasse ist Grundlage für die Erzeugung von Objekten, indem sie deren Attribute und Methoden in detail definiert, also ein Bauplan.
- Die Prototypvorstellung: Eine Klasse definiert zwar ebenfalls detailliert, welche Struktur Objekte haben und wie sie arbeiten sollen, wird aber prototypisch gedacht, was bedeutet, dass alle erzeugten Objekte Kopien der Klasse sind.

Die Mengen- und die Bauplanvorstellung wurden bereits in [Ra11] als Grundvorstellungen identifiziert, da sie den in Abschnitt 2 formulierten Anforderungen voll entsprechen. Anders als in [ET05] sehen wir diese Vorstellungen nicht als aufeinander aufbauend an. Vielmehr sind diese Grundvorstellungen in gewisser Weise als gleichwertig anzusehen. Die Vorstellungen sind unseres Erachtens nämlich in verschiedenen Kontexten verschieden wichtig: Bei der Modellierung einer Realsituation wird vorrangig die Mengenvorstellung benötigt (vgl. dazu [Br09], S. 135: „Ein erster Schritt in der Entwurfsphase ist die Identifizierung der in der Auftragsituation vorkommenden Objekte und ihre Zuordnung zu Klassen“), während in der Implementierungsphase die Bauplanvorstellung offensichtlich die hilfreichere ist, da der konstruktive Aspekt von Klassen in den Vordergrund tritt.

Die Prototypvorstellung ist der Bauplanvorstellung sehr nahe, widerspricht aber stark der Mengenvorstellung. Darüber hinaus impliziert diese Vorstellung, dass eine Klasse als Objekt, eben als Prototyp, neben den verwendbaren Objekten existiert. Deshalb muss noch weiter diskutiert werden, wie tragfähig diese Vorstellung ist und ob man sie als Fehlvorstellung bezeichnen sollte.

Nicht als Grundvorstellung identifiziert haben wir eine auf der Fabrikmetapher (vgl. [We07], S.118) basierende Vorstellung, da diese bezogen auf hilfreiche Aspekte wie die detaillierte Beschreibung der zu konstruierenden Objekte nicht signifikant von der Bauplanvorstellung abweicht. Außerdem beinhaltet diese Vorstellung den irreführenden Aspekt, dass eine Klasse (und nicht ein Konstruktoraufwurf) die entsprechenden Objekte konstruiert.

4 Analyse und Ergebnisse der Befragungen

	Klasse	Objekt	Klasse	Objekt
Kurs 1	Me	Me	Me	Bü
	Bü	Bü		
	Ba	Ba	Ba	Ba
	Bü	Ba	Bü	
	Ba	Ba	Ba	Ba
	Bü	Me	Bü	Me
	Bü	Ba	Ba	
	Me		Ba	Ba
	Bü		Ba	Ba
	Ba		Ba	Ba
			Ba	Ba
		Ba	Ba	Ba
Kurs 2	Me	Me		
	Ba	Ba	Ba	Me
			Bü	Bü
	Ba	Ba	Ba	
	Bü		Ba	
	Me	Me	Me	Me
	Me	Me		
			Ba	Ba
	Me		Me	
	Ba	Ba		
	Me	Me	Me	Bü
	Ba	Me	Bü	Me
		Bü	Ba	
	Bü			
Befragung 1		Befragung 2		

Tabelle 1- Übersicht der Befragungsergebnisse

Die in den Befragungen erhobenen Schülertexte wurden mit Mitteln der qualitativen Inhaltsanalyse (vgl. [F110]) untersucht. Durch die Kategorisierung der entsprechenden Formulierungen konnten in den Befragungen verschiedene Schülervorstellungen der Beziehung zwischen Klassen und Objekten identifiziert werden. In Tabelle 1 findet man eine codierte Darstellung der transkribierten Beschreibungen. Jeder Beschreibung wurde die als dominant identifizierte Vorstellung zugeordnet, während eine detaillierte Analyse auch Indizien für das Vorhandensein weiterer Vorstellungen lieferte.

Der Code *Ba* steht in der Codierung für die Bauplanvorstellung und *Me* für die Mengenvorstellung. Bei einigen wenigen Beschreibungen blieb auch nach mehreren Analysedurchgängen unklar, welche Vorstellungen ihnen zugrunde lagen. Die entsprechenden Zellen der Tabelle sind hier quer gestreift. Leere Zellen bedeuten, dass der Proband an der jeweiligen Befragung nicht teilgenommen hat.

Hinweise auf die Bauplanvorstellung waren zum einen sehr konkrete Formulierungen wie: „eine ‚Bauanleitung‘ für etwas, was konstruiert werden soll, sodass man viele z.B. Dreiecke unterschiedlich nach dieser Anleitung konstruieren kann.“ Zum anderen wurden auch Beschreibungen, die eher implizit auf die Bauplanvorstellung hinwiesen, dieser Kategorie zugeordnet: „Eine Klasse beinhaltet Eigenschaften, welche jedes Objekt in dieser Klasse besitzt, es **definiert** also das Objekt.“

Allerdings weist die Formulierung „in dieser Klasse“ innerhalb der letztgenannten Beschreibung ebenfalls auf das Vorhandensein der Mengenvorstellung hin. Eindeutigere Hinweise auf die Mengenvorstellung boten Formulierungen wie „Gruppierung mehrerer Objekte, kann unterschiedliche Objekte beinhalten“ und „Ein Objekt ist eine Variable, die einer Klasse zugeordnet wurde und die Attribute mit eingetragenen Werten besitzt.“

Erst mehrfache Textanalysen machten es möglich, einige zunächst nicht kategorisierbare Beschreibungen zu charakterisieren: Die mit *Bü* codierten Beschreibungen deuten auf eine Vorstellung hin, die nicht unter den postulierten zu finden war. Es ist angebracht, sie als Bündelungsvorstellung bezeichnen. Von den jeweiligen Probanden wurde in den Vordergrund gerückt, dass Klassen strukturierte Datentypen, also quasi ein "Bündel" von Attributen und Methoden, darstellen.

Wir vermuten, dass die im Unterricht umgesetzte Einführung von Klassen und Objekten über Datenstrukturen wie Listen oder Records zunächst zur Entwicklung dieser Vorstellung führt, die für das erste Verständnis zwar hilfreich, für die vertiefte Verwendung jedoch nur beschränkt nützlich ist, da sie den Unterschied zwischen Klasse und Objekt nicht zwingend erklärt und die Notwendigkeit der Objekterzeugung für sich allein genommen nicht begründet.

Naheliegend ist daher, dass ein Verharren bei der Bündelungsvorstellung von einem nur oberflächlichen Verständnis der Konzepte Klasse und Objekt zeugt, was dazu führt, dass die Lernenden sowohl in Modellierungs- als auch in Implementierungssituationen scheitern. Weitere Untersuchungen sollen zeigen, ob man diese Vermutung bestätigen kann oder verwerfen muss. Genaueres folgt im nächsten Abschnitt.

Bei nur einem Probanden wurde die postulierte Prototypvorstellung identifiziert. Die Identifikation ist außerdem lediglich schwach belegt durch das Wort „weiterzuverwenden“ in der Beschreibung „...Klassen ist nützlich, um ein Objekt für weitere Anwendungen weiterzuverwenden.“.

Wie bereits weiter oben erwähnt, geben viele der Beschreibungen Hinweise auf die Existenz mehrerer Vorstellungen bei dem jeweiligen Probanden. Die Auswahl in Tabelle 1 demonstriert dies beispielhaft an einer Probandin und einem Probanden:

Proband	Klasse	Objekt	Klasse	Objekt
36	Eine Klasse definiert einzelne Objekte. Sie ist im Prinzip eine "Bauanleitung" in der alle Eigenschaften beschrieben werden. Die Eigenschaften können verändert werden, so kann zum Beispiel die Länge und Breite eines Rechtecks verändern, die Bauanleitung ist aber auf jeden Fall für ein Rechteck.	Ein Objekt ist das, was von der Klasse konstruiert wird.	Eine Klasse definiert ein Objekt und ordnet Attribute und Methoden zu. Sie kann als "Bauanleitung" für ein Objekt angesehen werden.	Ein Objekt wird von einer Klasse definiert, ist also ein Bestandteil einer Klasse . Ihm werden innerhalb einer Klasse Attribute und Methoden zugeordnet, auf deren Grundlage es später vom Konstruktor erzeugt wird.
21	eine Sammlung von Attributen und Methoden , die zu einer Gruppe zusammengefasst sind. Alle Attribute sind ohne Wert, bzw. können einem Wort zugeordnet werden.	Ein Objekt ist eine Variable, die einer Klasse zugeordnet wurde und die Attribute mit eingetragenen Werten besitzt.	eine Sammlung von Attributen , die bestimmte Eigenschaften haben. Es gibt eine bestimmte Methoden, die angibt, wie man eine Klasse angibt.	ein Objekt kann einer Klasse zugeordnet werden. Ein Objekt kann genau einer K. zugeordnet werden. Ein Objekt besitzt definierte Eigenschaften, die Objekte untereinander unterscheiden

Tabelle 2 - Ausgewählte Beschreibungen

Die Beschreibungen von Probandin 36 offenbaren beispielsweise Indizien für die Bauplan- und die Mengenvorstellung: Die Bauplanvorstellung kommt explizit in der Beschreibung vor: Eine Klasse „ist im Prinzip eine Bauanleitung“, ein Objekt wird „von der Klasse konstruiert“ oder „vom Konstruktor erzeugt“.

Die Formulierung „Bestandteil einer Klasse“ in der Beschreibung von Objekten der zweiten Befragung bringt dagegen die Mengenvorstellung implizit zum Ausdruck. Hinweise auf die Bauplanvorstellung überwiegt allerdings an dieser Stelle, weshalb in der vergrößerten Ansicht (Tab. 1) für alle Einträge der vorrangige Code *Ba* gewählt wurde.

Interessant ist an dem Beispiel der Probandin 36 aus Tabelle 2 außerdem die Ausschärfung der Begriffe im Laufe der Zeit, in der die Schülerinnen und Schüler über ein halbes Jahr lang keinen OO-Unterricht hatten: Während bei der ersten Befragung ein Objekt noch als „von der Klasse konstruiert“ bezeichnet wird (vgl. „Fabrikmetapher“ [We07], S. 118), verwendet die Probandin in der zweiten Befragung die Worte „vom Konstruktor erzeugt“, was klar die Rolle des Konstruktors (als Erbauer) von der Rolle der Klasse (als Bauanleitung) trennt.

Den Beschreibungen von Proband 21 konnten sowohl die Bündelungsvorstellung („eine Sammlung von Attributen und Methoden“) sowie die Mengenvorstellung („ein Objekt kann einer Klasse zugeordnet werden“) entnommen werden. Weitere Untersuchungen müssen in ausgewählten Fällen klären, welche Grundvorstellungen die Probanden bezüglich Klassen und Objekten in Anwendungssituationen aktivieren können.

Insgesamt konnte in 13 Fällen die Bauplanvorstellung bereits in der ersten Befragung festgestellt werden, in weiteren sechs Fällen zeigte sich eine Entwicklung zur Bauplanvorstellung und nur ein Fall dokumentierte eine Entwicklung zur Mengenvorstellung. Die Weiterentwicklung zur Bauplanvorstellung bzw. eine Ausschärfung in diese Richtung konnte man insbesondere in Kurs 1 feststellen, in dessen Unterricht ein zusätzliches Projekt zur objektorientierten Modellierung und insbesondere zur objektorientierten Implementierung eines Computernetzwerks realisiert wurde.

Daraus kann man die Vermutung ableiten, dass mehr Erfahrung mit der Implementierung objektorientierter Modellierungen die Bauplanvorstellung stärker in den Vordergrund rückt. Kontrastierend könnte man behaupten, dass eine Schwerpunktsetzung auf Situationsmodellierung die Ausbildung der Mengenvorstellung befördert.

Desweiteren konnte festgestellt werden, dass einige Beschreibungen aus der ersten Befragung sehr nah an konkreten Inhalten des Unterrichts bzw. Analogien des Alltags waren, während die Äußerungen aus der zweiten Befragung ein höheres Abstraktionsniveau aufwiesen. Diese Beobachtung entspricht der Erkenntnis aus [Th02] bzw. [Du94], dass sich Modellvorstellungen besonders zu Beginn des Lernprozesses an Analogien zu Konzepten der Erfahrungswelt orientieren.

Beispielsweise wurde von einem Probanden in der ersten Befragung angegeben, dass man sich „eine Klasse wie einen Schrank vorstellen“ könne, während er in der zweiten Befragung beschrieb: „Eine class hat die Bauanleitung inne, mit der ein Objekt erstellt wird“. Auch hier war diese Tendenz zur Loslösung von konkreten Beispielen besonders in Kurs 1 ausgeprägt, dessen Mitglieder im Unterricht mit der Modellierung und Implementierung einer Simulation von Rechnernetzen befasst waren.

Über die bereits beschriebenen Erkenntnisse und Folgerungen hinaus konnten bezüglich der vorhandenen Grundvorstellungen keine genderspezifischen Unterschiede festgestellt werden.

5 Zusammenfassung und Ausblick

Zunächst stellen wir fest, dass die Erhebung von Schülervorstellungen erfreulich umfassend gelungen ist. Das sehen wir als Beleg dafür, dass die Identifikation von Grundvorstellungen wie im Vorfeld angenommen ein adäquates Mittel ist, wichtige Aspekte mentaler Modelle zu erheben. Das hängt selbstverständlich mit der Intention des Konzeptes zusammen: Grundvorstellungen versuchen, mentale Modelle zu beschreiben, indem sie wichtige Aspekte eines mentalen Modells kommunizierbar instanzieren. Dabei ist zu beachten, dass nach der Definition des Grundvorstellungsbegriffs, wie eingangs erwähnt, mehrere Grundvorstellungen zu einem mentalen Modell gehören, was auch die Analyse der durchgeführten Befragung belegt. Wenn ein Schüler oder eine Schülerin also in typischen Anwendungssituationen die benötigten Grundvorstellungen aktivieren könnte, könnte man sein oder ihr mentales Modell als tragfähig bezeichnen. Die drängendsten offenen Fragen sind also

- Können Schülerinnen und Schüler, deren Beschreibungen Grundvorstellungen informatischer Begriffe und Konzepte aufweisen, diese auch in Anwendungssituationen abrufen?
- Folgt aus dem Verharren bei der Bündelungsvorstellung das Scheitern in Anwendungssituationen?

Als weitere Untersuchungen sind Aufgabenanalysen und Interviews geplant. Zunächst werden dafür Aufgaben erstellt und auf ihren Grundvorstellungsgehalt hin untersucht. Mithilfe von auf diese Weise erstellten Aufgaben werden in der Folge videografierte Problemlösungen und Interviews konzipiert, durchgeführt und ausgewertet.

Ein langfristiges Ziel ist darüber hinaus die Entwicklung eines Unterrichtskonzepts, dessen Kern eine Variation der Kontexte und Problemstellungen entlang der Grundvorstellungsdimension im Sinne der „variation theory“ ([ET05], S. 90) sein soll. Zunächst soll noch untersucht werden, welche Kontexte und Problemstellungen für die Ausbildung geeigneter Grundvorstellungen günstig sind.

Literaturverzeichnis

- [BG10] Barry, P.; Griffiths, D.: Programmieren von Kopf bis Fuß. O'Reilly, Köln, 2010.
- [Br09] Brichzin, P. et al.: Informatik – Oberstufe 1, Datenstrukturen und Softwareentwicklung. Oldenbourg Schulbuchverlag, München, 2009.
- [CLB07] Caspersen, M.E., Larsen, K.D., Bennedsen, J.: Mental models and programming aptitude. In: Proceedings of the 12th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education. ITiCSE, ACM New York, 2007.

- [Du94] Dutke, S.: Mentale Modelle: Konstrukte des Wissens und Verstehens. Kognitionspsychologische Grundlagen für die Software-Ergonomie. Verlag für Angewandte Psychologie, Göttingen, 1994.
- [ES09] Ehlert, A. and Schulte, C.: Empirical comparison of objects-first and objects-later. In: Proceedings of the Fifth international Workshop on Computing Education Research Workshop, S.15-26, ICER '09, ACM, New York, NY, 2009.
- [ET05] Eckerdal, A.; Thuné, M.: Novice Java Programmers' Conceptions of "Object" and "Class", and Variation Theory. In: Proceedings ITiCSE'05, June 27-29, Monte de Caparica, Portugal, S. 89-93, 2005.
- [Fl10] Flick, U.: Qualitative Sozialforschung. 3. Aufl., Rowohlt Taschenbuch Verlag, Reinbek bei Hamburg, 2010.
- [Ho95] vom Hofe, R.: Grundvorstellungen mathematischer Inhalte. Spektrum Akademischer Verlag, Heidelberg, Berlin, Oxford, 1995.
- [Ko09] Kortenkamp, U. et.al.: Objektorientierte Modellierung - aber wann und wie? Zur Bedeutung der OOM im Informatikunterricht. LOG IN 29(160/161): S. 41-47, LOG IN-Verlag, Berlin, 2009.
- [Ma07] Ma, L.: Investigating and Improving Novice Programmers' Mental Models of Programming Concepts. Dissertation, University of Strathclyde, Glasgow 2007.
- [No83] Norman, D.A.: Some Observations on Mental Models. In: (Gentner, D.; Stevens, A.L. Hrsg.) Mental models. [7. pr.]. Erlbaum, Hillsdale, NJ, USA, 1983; S. 7-14
- [ORS12] Oldenburg, R.; Rabel, M.; Schuster, J.: "A Turtle's Genetic Path to Object Oriented Programming" in Proceedings to Constructionism, 21.-25.08. 2012 in Athen, The Educational Technology Lab, Department of Pedagogy, University of Athens, 2012.
- [Ra11] Rabel, M.: Grundvorstellungen in der Informatik. In: M. Weigend, M. Thomas, & F. Otte (Hrsg.), Informatik mit Kopf, Herz und Hand. Praxisbeiträge zur Infos 2011, S. 61–70. Münster: ZfL-Verlag.
- [Sc11] Schuster, J.: "Ein genetischer Zugang zum Programmieren mit CGI-Skripten in Python" in Marco Thomas (Hrsg.): Informatik in Bildung und Beruf, INFOS 2011 - 14. GI-Fachtagung Informatik und Schule, 12.-15. September 2011 in Münster, GI-Edition - Lecture Notes in Informatics (LNI), P-189, Bonner Köllen Verlag, 2011.
- [Th02] Thomas, M.: Informatische Modellbildung. Modellieren von Modellen als zentrales Element der Informatik für den allgemeinbildenden Schulunterricht. Dissertation, Universität Potsdam, Potsdam, 2002.
- [We07] Weigend, M.: Intuitive Modelle der Informatik. Dissertation, Universitätsverlag Potsdam, 2007.
- [Wi81] Wittmann, E.: Grundfragen des Mathematikunterrichts. 6. Aufl., Vieweg, Braunschweig, 1981.

Movingart – ein fächerübergreifendes Kunst-Informatik-Projekt in der Sekundarstufe I

Sönke Schulmeister

Institut für Informatik
Christian-Albrechts-Universität zu Kiel
Christian-Albrechts-Platz 4
24118 Kiel
schulmeister@ti.informatik.uni-kiel.de

Abstract: In diesem Artikel wird eine Studie über Movingart, ein über zwei Jahre in der Sekundarstufe I in Schleswig-Holstein durchgeführtes Kunst-Informatik-Projekt, vorgestellt. In der Studie wurde untersucht, ob die Schülerinnen und Schüler in einem überwiegend künstlerischen Kontext elementare Informatikkenntnisse erwerben. Es wurde ferner untersucht, von welchen Faktoren dieser Erwerb abhängig ist, welche Merkmale des Projektes besonders motivierend für die Schülerinnen und Schüler waren und ob Movingart zu einem verstärkten Interesse an der Informatik führt.

Es zeigt sich, dass die Schülerinnen und Schüler grundlegende Informatik- und Programmierkenntnisse, wie sie im ersten Jahr eines regulären Informatikunterrichts in der Sekundarstufe I vermittelt werden, erwerben. Der projekt- und produktorientierte Charakter von Movingart, der Wettbewerb und auch der fächerübergreifende Ansatz werden von den Schülerinnen und Schüler als motivationale Elemente benannt. Die Forschungsergebnisse legen die Einrichtung eines eigenen Kunst-Informatik-Wahlpflichtfachs nahe, um das geweckte Interesse an der Informatik und Kunst weiter vertiefen zu können.

1 Einleitung

Seit vielen Jahren verzaubert das Pixar-Animationsstudio die Zuschauer mit fantasievollen und grafisch atemberaubenden Filmen. Vielen Menschen ist dabei nicht bewusst, dass nicht nur Künstler, sondern gleichermaßen Informatiker an der Entstehung der Filme beteiligt sind. Ohne die Entwicklung immer ausgefeilterer Rendering- und Animationsalgorithmen könnten viele Ideen nicht umgesetzt werden. Auch bei professionellen Spieleproduktionen gibt es eine fruchtbare Symbiose aus Informatik und Kunst, um aus informatischer und künstlerischer Sicht ansprechende Ergebnisse zu liefern [Due04]. In der Schule sind hingegen Projekte, die Kunst- und Informatikinhalte gleichermaßen berücksichtigen, selten. Im Informatikunterricht reduzieren sich künstlerische Aspekte oft auf die Programmierung von Turtle-Grafiken oder bestenfalls Grafiken, die einem 3D-Raytracing-Programm entstammen. Die „Informatik“ im Kunstunterricht beschränkt sich zumeist auf das Erlernen der Handhabung gängiger Grafiksoftware [Deh01]. Durch diese Beobachtung motiviert, entstand im Institut für Informatik der Universität Kiel ein Projekt für

den Kunstunterricht der Sekundarstufe I in Schleswig-Holstein, das sowohl künstlerische als auch informatische Aspekte berücksichtigte - das Movingart-Projekt¹. Als Entwicklungswerkzeug wurde Scratch² gewählt. Ziel der Studie war es herauszufinden, ob die Schülerinnen und Schüler³ im überwiegend künstlerisch angelegten Projekt elementare Informatikkenntnisse erwerben.

Dieser Artikel beschreibt im Abschnitt 2 die genaue Konzeption des Movingart-Projektes, die Wahl der Programmierumgebung, die zur Verfügung gestellten Lernmaterialien und das mit dem Projekt einhergehende Fortbildungsangebot. Im Abschnitt 3 werden die Ziele der Untersuchung zum Movingart-Projekt dargestellt. Die Forschungsmethodik, Ergebnisse und eine Diskussion folgen in den Abschnitten 4 bis 6.

Aufgrund der vorgegebenen Beschränkung auf zehn Seiten, enthält diese Fassung nicht alle statistischen Ergebnisse. Diese finden sich, zusammen mit ausführlicheren Beschreibungen zur Methodik, in der Langfassung dieses Artikels, die unter <http://bit.ly/13DXmCi> einzusehen ist.

2 Das Movingart-Projekt im Überblick

2.1 Konzeption

Das Movingart-Projekt ist primär für den Kunstunterricht der Sekundarstufe I in Schleswig-Holstein entwickelt worden. Im Mittelpunkt standen künstlerische Inhalte (u.a. Kunst der Animation, Wirkung von Perspektiven, Storytelling). Dennoch wurden bei der Konzeption des Projektes elementare informatische Inhalte berücksichtigt (logisches Planen und Umsetzen von Abläufen unter Berücksichtigung elementarer Programmiertechniken (Fallunterscheidungen, Schleifen, Variablen)). Damit finden sich Teile des Movingart-Projektes sowohl im Lehrplan des Fachs Kunst [Min13b] als auch im Lehrplan des Fachs Angewandte Informatik [Min13a] in Schleswig-Holstein wieder. Hauptlernziel des Movingart-Projektes war es, dass die SuS lernen ihr eigenes Projekt (Animation oder Spiel) unter Berücksichtigung künstlerischer und informatischer Gesichtspunkte zu planen und zu verwirklichen. Insbesondere ist damit den Thesen von Prensky [Pre01] und Romeike [Rom07] Rechnung getragen worden, die von einer anderen Art des Lernens bei den „digital natives“ ausgehen – aktives statt passives Lernen, Nutzung unterschiedlicher Medien (Filme, Bilder, Klänge), kreatives Gestalten und Projektorientierung (Herstellung eines eigenen Produkts).

Ein weiteres Gestaltungsmerkmal des Projektes ist die Implementation eines Wettbewerbs, der, gestaffelt nach Altersgruppen, eine Beurteilung der von den SuS hergestellten Animationen und Spielen durch eine Jury vorsah. Die SuS haben diese Beiträge im Anschluss an die erste Unterrichtsphase, in der die künstlerischen und technischen Grundlagen für die Umsetzung eigener Projekte gelegt worden sind, in einer freien Projektarbeit erstellt.

¹Vgl. auch die aktuelle STEAM-Diskussion in den USA: <http://bit.ly/142QYqM>

²<http://scratch.mit.edu/>

³Im weiteren Verlauf mit SuS abgekürzt.

Im Rahmen der Software Challenge [Ins13a], einem Programmierwettbewerb für die Sekundarstufe II, hat das Institut für Informatik in Kiel bereits gute Erfahrungen hinsichtlich der Motivation der teilnehmenden SuS, bedingt durch den Wettkampfcharakter, gemacht. Auf einer Abschlussveranstaltung, zu der alle SuS eingeladen waren, die am Movingart-Projekt teilgenommen haben, wurden die ausgezeichneten Beiträge vorgestellt und Preise vergeben. Das Movingart-Projekt fand in zwei Durchgängen mit jeweils anderen SuS statt (1. Durchgang: August 2010 - Februar 2011 (387 SuS), 2. Durchgang: August 2011 - Februar 2012 (348 SuS)). In beiden Durchgängen standen für die erste Unterrichtsphase und die sich anschließende Wettbewerbsphase ca. 20 Schulwochen mit jeweils 1-2⁴ Schulwochenstunden zur Verfügung.

2.2 Scratch als Programmierumgebung

In den letzten Jahren sind viele Programmierumgebungen geschaffen worden, um in spielerischer Art mediale Projekte zu verwirklichen und elementare Kenntnisse der Informatik zu vermitteln. Für das Movingart-Projekt wurde Scratch als Werkzeug zur Mediengestaltung gewählt, weil es für die Zielgruppe (5. - 10. Klasse) durch die visuelle Programmiersprache und durch ihre klar strukturierte, didaktisch reduzierte Entwicklungsumgebung besonders geeignet erscheint, eigene Animationen und Spiele zu entwickeln ([Res09], [Fin10]). Positive Erfahrungen mit Scratch bei der Vermittlung informatischer Konzepte hat man sowohl in Nachmittagskursen für Jugendliche [Mal08], in Kursen der Sekundarstufe I [Mee10] als auch an Universitäten [Rom10] gemacht.

2.3 Lernmaterialien

Für die Entwicklungsumgebung Scratch stand zu Beginn des Movingart-Projekts keine deutschsprachige Literatur zur Verfügung. Insbesondere gab es keine detaillierten Unterrichtsmaterialien, um die Kunstlehrkräfte im Unterricht bei der Einführung in Scratch zu unterstützen, wie es die erste Unterrichtsphase vorsah. Aus diesem Grund wurden in Zusammenarbeit mit Informatik- und Kunstlehrkräften eigene Lernmaterialien für den Unterricht entwickelt, die zum einen ausführliche didaktisch-methodische Hinweise für die Lehrkräfte enthielten und zum anderen Arbeitsmaterialien, die es den SuS erlaubten, Scratch so zu nutzen, dass später in der Wettbewerbsphase auch komplexere Projekte umgesetzt werden konnten⁵.

⁴Abhängig von Schule und Klasse.

⁵<http://kunst-trifft-informatik.uni-kiel.de/material/>

2.4 Fortbildungen und Hospitationen

Da das Movingart-Projekt hauptsächlich im Kunstunterricht (und nur in wenigen Projektkursen) durchgeführt worden ist, wurden die teilnehmenden Kunstlehrkräfte, die fast alle keine informatische Ausbildung absolviert haben, vorab in Workshops im Umgang mit Scratch und den zur Verfügung gestellten Lernmaterialien geschult. Während des Unterrichts hospitierten studentische Hilfskräfte bei Bedarf den Kunstlehrkräften, um sie bei der Vermittlung informatischer Inhalte zu unterstützen.

3 Forschungsfragen

Aus vorherigen Studien wissen wir bereits, dass Scratch ein geeignetes Werkzeug darstellt, um SuS in die Programmierung einzuführen. Diesen Studien lag jedoch ein reiner Informatikunterricht zugrunde, in dem der Fokus primär auf der Vermittlung von Programmier-techniken lag [Wol09], [Mee10]. In dieser Studie soll hingegen der bisher noch nicht untersuchten Frage nachgegangen werden, ob elementare Informatikkenntnisse auch dann erworben werden, wenn eine Auseinandersetzung mit Scratch im Kunstunterricht und unter überwiegend künstlerischen Gesichtspunkten stattfindet.

Die Forschungsfragen sind im Einzelnen: (F1) Erwerben SuS, die am Movingart-Projekt teilnehmen, elementare Informatikkenntnisse? (F2) Ist ein Erwerb von Informatikkenntnissen abhängig vom Grad der Vorerfahrungen mit dem Computer, von der schulischen Leistungsstärke, vom Klassenjahrgang und vom Geschlecht? (F3) In welchen Bereichen erkennen SuS für sich selbst einen Lernzuwachs? (F4) Stellen die Herstellung eines eigenen medialen Produkts, der Wettbewerb und der fächerübergreifende Charakter des Projekts motivierende Elemente im Unterricht dar? (F5) Führt die Teilnahme am Movingart-Projekt zu einem verstärkten Interesse an Scratch und an informatischen Themen?

4 Methodik

4.1 Teilnehmerinnen und Teilnehmer der Untersuchung

An der Untersuchung nahmen in beiden Jahren 260 SuS der Jahrgangsstufen 7 bis 10 teil. Diese Anzahl unterteilt sich in zwei Forschungsgruppen: Den Teilnehmerinnen und Teilnehmern am Movingart-Projekt (M-Gr). Diese Gruppe umfasst 144 SuS. Den SuS, die nicht am Movingart-Projekt teilgenommen haben, aber aus den jeweils gleichen Jahrgängen und Schulen der Movingart-Teilnehmer stammen. Sie bilden die Kontrollgruppe (K-Gr) und umfassen 116 SuS.

4.2 Forschungsinstrumente

Für die Untersuchung der oben formulierten Fragen wurden ein Vor- und Nachtest und ein Fragebogen eingesetzt.

4.2.1 Vor- und Nachtest

Sowohl Vor- als auch Nachtest umfassten jeweils zehn Aufgaben. Die ersten sieben Aufgaben hatten jeweils Multiple-Choice-Format, die letzten drei Aufgaben waren offen gestaltet und gestatteten individuelle Lösungen der SuS. Der Nachtest wies neue Aufgaben auf, orientierte sich aber an den Aufgabenstellungen und -inhalten des Vortests. Der Vortest wurde in beiden Jahren von beiden Forschungsgruppen ((M-Gr) und (K-Gr)), zu Beginn des Movingart-Projekts bearbeitet. Die Bearbeitung des Nachtests erfolgte dann ebenfalls von beiden Forschungsgruppen nach Abschluss des Movingart-Projekts. Die Form der Testaufgaben lehnte sich an den Aufgaben des Informatik-Biber-Wettbewerbs an [Poh13].

4.2.2 Fragebogen

Am Ende des Movingart-Projekts füllten alle SuS des Movingart-Projekts, jeweils unter Anleitung der Lehrkraft, einen Fragebogen aus. Der Fragebogen umfasste sechs Teilbereiche: Allgemeine Fragen (8 Items), Fragen zur bisherigen Computernutzung (15 Items), allgemeine Fragen zum Movingart-Projekt (14 Items), spezielle Fragen zum Movingart-Projekt (8 Items), Fragen zum selbst festgestellten Lernerfolg (14 Items) und Fragen zur zukünftigen Gestaltung des Movingart-Projekts (4 Items).

4.3 Teilgruppen

Mit Hilfe eines Codes, der auf beiden Tests und dem Fragebogen einzutragen war, konnten Vor- und Nachtest und der Fragebogen genau einer Schülerin bzw. einem Schüler zugeordnet werden. Insgesamt konnten in beiden Jahren zusammen $N = 260$ Zuordnungen vorgenommen werden. Aus den erhobenen Daten wurden neben einer geschlechtsspezifischen Unterteilung sechs neue Teilgruppen gebildet: Die Gruppen der computervorerfahreneren (C^+) und computerunerfahreneren (C^-) SuS, die Gruppen der in der Schule leistungsstärkeren (L^+) und leistungsschwächeren (L^-) SuS und die Gruppen der jüngeren Jahrgänge ($J^- = 7.$ und 8. Klasse) und der älteren Jahrgänge ($J^+ = 9.$ und 10. Klasse).

5 Ergebnisse

5.1 F1: Erwerben SuS, die am Movingart-Projekt teilnehmen, elementare Informatikkenntnisse?

Ein für den Vor- und Nachtest gerechneter t -Test für unabhängige Stichproben ergibt: Die Ergebnisse des Vortests unterscheiden sich nicht signifikant, der Mittelwertsunterschied ist auf dem zugrunde gelegten Signifikanzniveau von 5% zufällig ($t(230) = -.9929; p = .3218 > .05$). Anders beim Nachtest: der t -Test ist mit $t(255) = -10.571; p \leq .01$ signifikant, d.h. die SuS der Movingart-Gruppe erzielten im Mittel signifikant bessere Ergebnisse als die SuS der Kontrollgruppe.

5.2 F2: Ist ein Erwerb von Informatikkenntnissen gruppenspezifisch unterschiedlich?

Sowohl für den Vor- und Nachtest und allen Teilgruppen wurde ein t -Test für unabhängige Stichproben gerechnet. Es ergibt sich, dass beim Vortest die Gruppe J^+ einen signifikant höheren Punktedurchschnitt aufweist als die Gruppe J^- ($t(142) = -2.98; p \leq .01$). Beim Nachtest hingegen gibt es nur einen marginalen signifikanten Unterschied zwischen den Teilgruppen J^+ und J^- ($t(139) = -1.78; p = .07$). Auch die Jungen erzielten ein beim Vortest signifikant höheren Punktedurchschnitt als die Mädchen ($t(142) = 2.08; p \leq .05$). Dies gilt auch beim Nachtest ($t(139) = 3.74; p \leq .01$). Alle anderen Vergleiche zwischen den Teilgruppen weisen keine signifikanten Unterschiede im Punktemittelwert auf.

5.3 F3: In welchen Bereichen erkennen SuS für sich selbst einen Lernzuwachs?

Für die Teilbereiche Elementare Programmier Techniken, Grafik- und Audibearbeitung, Erstellung einer eigenen Animation und Erstellung eines eigenen Spiels sollten die SuS ihre Kompetenz auf einer Skala zwischen 1 (sehr gut) und 6 (gar nicht) für den Zeitpunkt vor dem Projekt und nach dem Projekt einschätzen. Für alle Teilbereiche ist ein t -Test für abhängige Stichproben, ein Wilcoxon-Test, gerechnet worden. Er zeigt, dass in allen vier Bereichen der selbst eingeschätzte Lernzuwachs signifikant ist (Elementare Programmier Techniken $V = 2831.5; p \leq .01$, Grafik- und Audibearbeitung $V = 1934; p \leq .01$, Erstellung einer Animation $V = 5696.5; p \leq .01$ und eines Spiels $V = 4520; p \leq .01$).

5.4 F4: Stellen die Herstellung eines eigenen medialen Produkts, der Wettbewerb und der fächerübergreifende Charakter des Projekts motivierende Elemente im Unterricht dar?

Die SuS wurden befragt, welche der charakteristischen Merkmale des Movingart-Projekts (Herstellung eines eigenen medialen Produkts (1), Wettbewerb (2) sowie fächerübergreifender Charakter des Projekts (3)) sie motivierend für ihre Mitarbeit im Unterricht empfanden. Um zu erkennen, ob es signifikante Unterschiede bezüglich der drei Merkmale gibt, wurde jeweils ein χ^2 -Test gerechnet. Alle drei Merkmale sehen signifikant mehr SuS als motivierende Elemente im Unterricht an ((1) $\chi^2 = 57.04, p \leq .01$; (2) $\chi^2 = 20.83, p \leq .01$; (3) $\chi^2 = 17.86, p \leq .01$). Im Vergleich der einzelnen Teilgruppen ergibt sich folgendes Bild: Der fächerübergreifende Charakter ist für signifikant mehr SuS der jüngeren Jahrgänge als der älteren Jahrgänge ein motivierendes Element im Unterricht ($\chi^2 = 6.05, p = .014 \leq .05$). Das gleiche Merkmal ist für Jungen signifikant motivierender als für Mädchen ($\chi^2 = 4.38, p = .036 \leq .05$). Für alle anderen Merkmale gibt es jeweils zwischen den Teilgruppen keine signifikanten Unterschiede.

5.5 F5: Führt die Teilnahme am Movingart-Projekt zu einem verstärkten Interesse an Scratch und informatischen Themen?

Im Fragebogen sollten die SuS des Movingart-Projekts die Fragen beantworten, ob die Teilnahme am Movingart-Projekt zu einem stärkerem Interesse für Informatik (1) und einer weiteren, über den Unterricht hinausgehenden, Beschäftigung mit Scratch (2) führt. Ein χ^2 -Test zeigt, dass sich signifikant mehr SuS mit Scratch weiterbeschäftigen wollen ($\chi^2 = 13.44, p \leq .01$). Ein Vergleich der Teilgruppen ergibt: Signifikant mehr jüngere SuS interessieren sich nun für informatische Themen als die älteren SuS ($\chi^2 = 6.82, p \leq .01$). Dies gilt auch für die Weiterbeschäftigung mit Scratch ($\chi^2 = 16.09, p \leq .01$). Ansonsten bestehen zwischen den Teilgruppen jeweils keine signifikanten Unterschiede.

6 Diskussion

Das Movingart-Projekt ist primär für den *Kunst*unterricht der Sekundarstufe I konzipiert und durchgeführt wurden. Dennoch beinhaltet das Projekt, bedingt durch seinen fächerübergreifenden Charakter, Teile, die im Informatik-Anfangsunterricht der Sekundarstufe I Thema sind. Es ist eine zentrale Fragestellung dieser Studie, ob sich auch im Rahmen des Kunstunterrichts, quasi nebenher, elementare informatische Kompetenzen konstituieren. Die Ergebnisse 5.1 zeigen, dass dies der Fall ist. Während sich die Ergebnisse des Vortests bei der Movingart- und Kontrollgruppe kaum voneinander unterscheiden, gibt es große Unterschiede bei den Ergebnissen des Nachtests zwischen beiden Gruppen. Die durchschnittlich erreichte Punktzahl ist in der Movingart-Gruppe um über 50% gestiegen. Dies ist im Vergleich zur Kontroll-Gruppe, die nur einen prozentualen Anstieg von 9% aufweist,

deutlich. Es scheint somit, als ob die notwendige Auseinandersetzung mit Anweisungen, Schleifen, Bedingungen etc. in Scratch, um im künstlerischen Zusammenhang Animationen und Spiele zu erstellen, zu einem besseren algorithmischen Verständnis führt. Es fällt einfacher, formale Darstellungen von Programmen zu verstehen, selbst welche zu entwerfen und in einfachere Programme umzusetzen. Die grundlegenden Programmier Techniken sind damit erworben worden. Ein Übergang zu einer komplexeren Programmiersprache (z.B. Java in der Sekundarstufe II) sollte als Konsequenz für die SuS deutlich leichter fallen. Dies zeigt auch eine Studie von Malan und Leitner [Mal07].

Die Untersuchungen in 5.2 zeigen zwar, dass die Jungen sowohl im Vor- als auch Nachtest einen signifikant höheren Punktedurchschnitt aufweisen, der prozentuale Anstieg des Punktemittelwertes bei den Mädchen (+48,44%) liegt jedoch nur knapp unter dem der Jungen (+51,6%). Dies belegt, dass die SuS geschlechtsunabhängig am Lernzuwachs partizipieren. Vielleicht ist es im Projekt gelungen, vermeintlich abschreckende technische Aspekte in einem künstlerischen Kontext anzubieten und so Vorbehalte und Ängste, die kontraproduktiv beim Lernen wirken können, gerade bei den Mädchen zu vermeiden. Kein großer Unterschied bestehen zwischen der Gruppe der computereffahreneren (+53,78%) und computerunerfahreneren (+49,93%) SuS. Dies mag auf den ersten Blick verwundern, erwartet man vielleicht von den computeraffinen SuS einen höheren Lernzuwachs. Berücksichtigt man jedoch, dass die meiste Freizeit mit dem Computer für konsumtive Zwecke genutzt wird [Med13], so relativiert sich die Vermutung. Die größten Unterschiede im Lernzuwachs gibt es zwischen den jüngeren (+56,12%) und älteren Jahrgängen (+42,22%). Unter Umständen liegt dies an den Lehrplänen anderer Fächer. Gerade im Fach Mathematik werden ab dem 8. Jahrgang öfter algorithmische Sichtweisen auf mathematische Themen eingenommen. Dies ist in den jüngeren Jahrgängen nicht der Fall, sodass sich eine weitgehend neue Auseinandersetzung mit informatischen Themen auch in einem höheren Lernzuwachs niederschlägt.

Die Ergebnisse 5.3 belegen, dass die SuS in allen Bereichen, die im Movingart-Projekt thematisiert worden sind, einen Lernzuwachs für sich erkennen. Damit gehen die subjektiven Selbsteinschätzungen mit den objektiven Testergebnissen (F1) einher. Besonders große Veränderungen sehen die SuS in ihrer Kompetenz, eine eigene Animation ($M_{vorher} = 4.2$; $M_{nachher} = 2.4$) oder ein eigenes Spiel ($M_{vorher} = 4.9$; $M_{nachher} = 3.3$) zu erstellen. Es scheint, als ob damit ein wesentliches Ziel des Movingart-Projekts erreicht worden ist: SuS zu befähigen, den Computer produktiv einzusetzen und eigene digitale Projekte zu erstellen. Auch in den beiden anderen Teilbereichen, elementare Programmier Techniken ($M_{vorher} = 4.2$; $M_{nachher} = 3.2$) und Grafik- und Audibearbeitung ($M_{vorher} = 3.6$; $M_{nachher} = 2.8$), erkennen die SuS für sich neu erworbene Kompetenz. Diese Ergebnisse bestätigen somit andere Studien ([Mee10], [Mos04]). Die Hoffnung eine intrinsisch motivierte Kreativität zu wecken, die SuS aus der Konsumentenrolle digitaler Medien zu befreien, scheint geglückt.

Drei wesentliche Gestaltungsmerkmale des Movingart-Projekts sind der fächerübergreifende Ansatz, der Wettbewerb und die Herstellung eines eigenen Spiels oder einer eigenen Animation. Die Ergebnisse 5.4 zeigen in der Betrachtung aller SuS, dass deutlich mehr als drei Viertel die Herstellung eines eigenen Produkts als motivierend für den Unterricht empfanden. Aber auch der Wettbewerbscharakter und der fächerübergreifende Ansatz des

Movingart-Projekts wurden mit je fast 70% als motivierend benannt. Die positiven Wettbewerbserfahrungen, die das Institut für Informatik im Rahmen der Software-Challenge gemacht hat, bestätigen sich auch hier. Im klassischen Unterricht steht oftmals das „Lösen eines Lehrerproblems“ im Vordergrund, was von SuS in der Regel als künstlich aufgesetzt und wenig motivierend empfunden wird. Hier erkennt man, dass der kreative Wettbewerb und die produktorientierte Unterrichtsausrichtung sehr motivierend wirken. Dies scheinen wichtige Gründe für den Lernerfolg zu sein, der oben beschrieben worden ist. Auch andere Studien ([Ber05], [Ric04]) belegen dies. Kritischer wird der fächerübergreifende Ansatz aus Sicht der Mädchen und der älteren Jahrgänge gesehen. Unter Umständen hängt dies mit der Implementierung des Movingart-Projekts im Kunstunterricht zusammen. In den Fragebögen antworten 48% der Mädchen, dass der Kunstunterricht zu viele informatische Aspekte aufweise. Bei den Jungen sind es nur 29%. Dies lässt jedoch nicht auf eine grundsätzlich ablehnende Haltung der Mädchen gegenüber informatischer Inhalte schließen, wie es auch die Daten von F5 zeigen. Vielmehr stellt sich die Frage, ob der Kunstunterricht der richtige Rahmen für das Movingart-Projekt ist. Unter Umständen ist ein eigenständiges Fach besser geeignet, das Informatik und Kunst zu gleichen Teilen beherbergt. Dazu unten mehr.

Die Ergebnisse 5.5 zeigen, dass ein stärkeres Interesse für informatische Themen geweckt worden ist. Dies trifft für fast die Hälfte aller SuS zu. Fast zwei Drittel aller SuS wollen sich sogar weitergehend über den Unterricht hinaus mit Scratch beschäftigen. Besonders ansprechend scheint Scratch für die jüngeren Jahrgänge zu sein. Dies deckt sich mit den Beobachtungen, die die Entwickler von Scratch in Nachmittagsprogrammierkursen gemacht haben ([Mon08] und [Mal08]). Eine besondere Rolle spielt hier wohl der intuitive, nicht technische Ansatz von Scratch, der gerade die jüngeren SuS zum Experimentieren und Entdecken einlädt.

Fast 80% der SuS würden gerne wieder am Projekt teilnehmen. Die im Rahmen des Wettbewerbs eingereichten Animationen und Spiele⁶ zeugen vom Spaß der SuS, die ohne eine große Motivation in dieser Qualität nicht entstanden wären. Die Wettbewerbsbeiträge überzeugten die Jury sowohl aus technischer als auch künstlerischer Sicht. Es scheint damit in diesem Projekt gelungen zu sein, Informatik und Kunst zu verbinden – beide Fächer haben voneinander profitiert.

Eine Fortsetzung der Fächerverbindung aus Kunst und Informatik ist sinnvoll. In welchem Rahmen könnte dies geschehen? In Schleswig-Holstein ist ein Curriculum [Ins13b] für einen zweijährigen Kunst-Informatik-Wahlpflichtunterricht der Sekundarstufe I entstanden. Der Wahlpflichtunterricht integriert die Fächer Kunst und Informatik gleichberechtigt in einem neuen und eigenständigen Unterrichtsangebot. Unter Umständen wird dadurch die Angst von den jeweiligen Fachlehrkräften, aber auch SuS (s.o.), genommen, zu viele Fachinhalte fächerübergreifenden Projekten opfern zu müssen, wenn diese im Fachunterricht direkt, also im Falle von Movingart im Kunstunterricht, stattfinden. Der Kunst-Informatik-Wahlpflichtunterricht begann ab dem Schuljahr 2012/2013. Hier bieten sich für die Zukunft weitere unterrichtsbegleitende Studien an, insbesondere ob eine systematische Informatikbildung (teilweise) ersetzt werden kann.

⁶Diese können hier eingesehen werden:

<http://kunst-trifft-informatik.uni-kiel.de/der-wettbewerb/>

Literatur

- [Ber05] Bergin, S.; Reilly, R. The influence of motivation and comfort level on learning to program. In *PPIG 2005: Proceedings of the 17th Workshop of the Psychology of Programming Interest Group*, Seiten 293–304, 2005.
- [Deh01] Dehne, B. Computer im Kunstunterricht. *LOG IN*, 21:38–48, 2001.
- [Due04] Duesing, J.; Hodgins, J. Team teaching animation art and technology. In *ACM SIGGRAPH 2004 Educators program, SIGGRAPH '04*, Seiten 4–, New York, NY, USA, 2004. ACM.
- [Fin10] Fincher, S.; Cooper, S.; Kölling, M.; Maloney, J. Comparing alice, greenfoot & scratch. In *SIGCSE '10: Proceedings of the 41st ACM technical symposium on Computer science education*, Seiten 192–193, New York, NY, USA, 2010. ACM.
- [Ins13a] Institut für Informatik, Christian-Albrechts-Universität zu Kiel. Software Challenge. <http://www.informatik.uni-kiel.de/software-challenge/2013/>, [Stand: Februar 2013].
- [Ins13b] Institut für Qualitätsentwicklung an Schulen Schleswig-Holstein. Fachcurriculum für ein Wahlpflichtangebot Kunst und Informatik. <http://faecher.lernnetz.de/faecherportal/dokumente/1329322768.pdf/>, [Stand: Januar 2013].
- [Mal07] Malan, D.; Leitner, H. Scratch for budding computer scientists. *SIGCSE Bull.*, 39:223–227, March 2007.
- [Mal08] Maloney, J.; Pepler, K.; Kafai, Y.; Resnick, M.; Rusk, N. Programming by choice: urban youth learning programming with scratch. *SIGCSE Bull.*, 40:367–371, March 2008.
- [Med13] Medienpädagogischer Forschungsverbund Südwest. JIM 2011–Basisuntersuchung zum Medienumgang 12- bis 19-Jähriger. <http://www.mpfs.de/index.php?id=225>, [Stand: Januar 2013].
- [Mee10] Meerbaum-Salant, O.; Armoni, M.; Ben-Ari, M. Learning computer science concepts with scratch. In *Proceedings of the Sixth international workshop on Computing education research, ICER '10*, Seiten 69–76, New York, NY, USA, 2010. ACM.
- [Min13a] Ministerium für Bildung und Kultur des Landes Schleswig-Holstein. Fachlehrplan Angewandte Informatik. <http://lehrplan.lernnetz.de/index.php?DownloadID=444>, [Stand: Februar 2013].
- [Min13b] Ministerium für Bildung und Kultur des Landes Schleswig-Holstein. Fachlehrplan Kunst. <http://lehrplan.lernnetz.de/index.php?DownloadID=30>, [Stand: Februar 2013].
- [Mon08] Monroy-Hernández, A.; Resnick, M. FEATURE: Empowering kids to create and share programmable media. *interactions*, 15(2):50–53, März 2008.
- [Mos04] Moskal, B.; Lurie, D.; Cooper, S. Evaluating the effectiveness of a new instructional approach. *SIGCSE Bull.*, 36(1):75–79, März 2004.
- [Poh13] Pohl, Wolfgang. Informatik-Biber-Wettbewerb. <http://informatik-biber.de/>, [Stand: Januar 2013].
- [Pre01] Prensky, M. Digital Natives, Digital Immigrants, Part II: Do They Really Think Differently? *On the Horizon*, Vol. 9, No. 6, 2001.
- [Res09] Resnick, M.; Maloney, J.; Monroy-Hernández, A.; Rusk, N.; Eastmond, E.; Brennan, K.; Millner, A.; Rosenbaum, E.; Silver, J.; Silverman, B.; Kafai, Y. Scratch: programming for all. *Commun. ACM*, 52:60–67, November 2009.
- [Ric04] Rich, L.; Perry, H.; Guzdial, M. A CS1 course designed to address interests of women. *SIGCSE Bull.*, 36(1):190–194, März 2004.
- [Rom07] Romeike, R. Kriterien kreativen Informatikunterrichts. In Sigrid E. Schubert, Hrsg., *INFOS*, Jgg. 112 of *LNI*, Seiten 57–68. GI, 2007.
- [Rom10] Romeike, R. Scratch2Greenfoot - Eine kreative Einführung in die Programmierung mit Scratch und Greenfoot. In *Didaktik der Informatik: Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik, ICER '10*, Seiten 107–112, Oldenburg, Germany, 2010. GI.
- [Wol09] Wolz, U.; Leitner, H.; Malan, D.; Maloney, J. Starting with scratch in CS 1. *SIGCSE Bull.*, 41(1):2–3, März 2009.

Informatische Kompetenzen im Gebiet „Datenbanken“ – eine Lernstandsanalyse in Sachsen und Thüringen

Büttner, K.; Damnik, G.; Friedrich, S.; Knapp, T.; Rätz, M.; Rohland, H.

TU Dresden, Fakultät Informatik
Arbeitsgruppe Didaktik der Informatik/Lehrerbildung
01602 Dresden
steffen.friedrich@tu-dresden.de

Abstract: Der vorliegende Beitrag stellt Planung, Durchführung und Ergebnisse einer Lernstandsanalyse im Fach Informatik in Sachsen dar. Dabei wird vor allem ein Workflow beschrieben, der für Untersuchungen von Kompetenzen bei Schülerinnen und Schülern unterschiedlicher Fächer und Klassenstufen prinzipiell geeignet sein sollte. Die Erfahrungen aus den Jahren 2008 und 2012, einschließlich der dabei vorgenommenen Modifizierungen des Ablaufs, sowie die jeweils hohe Anzahl der beteiligten Schulen und Schülerinnen und Schüler zeigen, dass hier ein für die Unterrichtspraxis realisierbares Vorgehen beschrrieben wurde. Sie verweisen gleichzeitig auch auf Möglichkeiten einer engen Zusammenarbeit zwischen Schule und Universität.

0. Vorwort

Basierend auf den Erfahrungen einer Lernstandsanalyse (LSA) im Jahre 2008 zum Themengebiet „Klasse – Objekt – Attribut – Methode“ [BKF09] konstituierte sich im September 2011 eine Arbeitsgruppe¹ zur Vorbereitung einer neuen LSA im Pflichtfach Informatik in Sachsen. Es galt ein Themengebiet zu finden, das (gegenüber der LSA aus dem Jahre 2008) ein höheres Schulalter berücksichtigt und in dem die zu erwartenden Leistungen bei Schülern sowohl der Mittelschule als auch des Gymnasiums entsprechend der Lehrplanvorgaben annähernd übereinstimmen. Das Thema „Datenbanken in der Sekundarstufe I“ erfüllte diese Anforderungen. Bei der Bewertung der Ergebnisse ist zu beachten, dass mit der Genehmigung der Lernstandsanalyse seitens des Kultusministeriums die Freiwilligkeit der Teilnahme der Schulen gefordert wurde.

1. Informatische Kompetenzen im Themengebiet Datenbanken

Ausgehend von der Auswahl des Themengebietes waren in einem ersten Schritt der Vorbereitung der LSA die Kompetenzen zu bestimmen, die einer Analyse unterzogen werden sollten. Die vorliegenden Grundsätze und Standards für die Informatik in der Schule (Bildungsstandards Informatik für die Sekundarstufe I) [BS07] lieferten eine geeignete Basis für diese Analyse. Aussagen zum Themengebiet lieferten dabei sowohl

¹ An der Arbeit zur LSA 2012 waren aktiv beteiligt: Büttner, K. Heidenau; Bollwig, C. Dresden; Damnik, G. Dresden; Drews, D. Bad Berka; Friedrich, S. Dresden; Fritzsche, H. Cunewalde; Knapp, T. Radebeul; Pampel, J. Lichtentanne; Rätz, M. Meißen; Rohland, H. Dresden; Roth C. Plauen; Thoss, R. Hainichen; Weiser C. Hohenstein-Ernstthal; Werner R. Chemnitz; Zuschke, S. Dresden

die Inhaltsbereiche „Information und Daten“ und „Informatik, Mensch und Gesellschaft“ als auch der Prozessbereich „Modellieren und Implementieren“. Folgende Kompetenzen sind dabei insbesondere in den Fokus der LSA gerückt:

Schülerinnen und Schüler aller Jahrgangsstufen

- nehmen Entscheidungsfreiheiten im Umgang mit Informatiksystemen wahr und handeln in Übereinstimmung mit gesellschaftlichen Normen,
- reagieren angemessen auf Risiken bei der Nutzung von Informatiksystemen,
- verstehen den Zusammenhang von Information und Daten sowie verschiedene Darstellungsformen für Daten,
- verstehen Operationen auf Daten und interpretieren diese in Bezug auf die dargestellte Information,
- führen Operationen auf Daten sachgerecht durch,
- erstellen informatische Modelle zu gegebenen Sachverhalten,
- reflektieren Modelle und deren Implementierung.

Anschließend musste geprüft werden, inwieweit die Anforderungen sächsischer Lehrpläne mit diesen Kompetenzfeldern übereinstimmen. Das ist einmal in der Mittelschule, Klasse 9, Lernbereich 1: "Daten modellieren: Datenbanksysteme" der Fall. Die verbindlichen Lehrplanforderungen sind dort [LP04] in Lernzielebenen (hier: kursiv dargestellt) formuliert und mit weitergehenden Forderungen unterlegt. Im Lehrplan des Gymnasiums [LP04] sind adäquate Inhalte enthalten. Diese befinden sich im Lehrplan des Profilbereichs der Klassenstufen 9/10, da die informatischen Inhalte dort integrativ unterrichtet werden.

Die für die LSA wesentlichen Lehrplanforderungen lassen sich so zusammenfassen:

- *Einblick gewinnen* in die Notwendigkeit des Datenschutzes
- *Kennen* der Strukturierung von Daten, Planen von Datenbanken, Darstellen von Beziehungen zwischen Tabellen in geeigneter Form
- *Beherrschen* des Umgangs mit Datensammlungen zur Informationsbeschaffung, Auswerten durch Sortieren und Filtern, Gewinnen neuer Informationen durch Abfragen

Letztendlich war das Fachgebiet in fachliche Bereiche zu zerlegen und von den Anforderungen der Sekundarstufe abzugrenzen. Hierzu ist ein Modell erarbeitet worden [Bo11], das eine wesentliche Grundlage der Aufgabenentwicklung darstellte und als permanentes Planungs- und Arbeitswerkzeug diente.

2. Etappen der Durchführung der LSA Informatik

Bei der Erstellung einer Lernstandsanalyse mit Schülern müssen einige wichtige Hürden genommen werden, die sowohl eine genehmigte Durchführung als auch einen für alle Beteiligten akzeptablen Ablauf garantieren. Das erfordert neben qualitativ und testtheoretisch hochwertigen Aufgaben vor allem auch die Sicherung einer organisatorischen und technischen Infrastruktur. Nach den Erfahrungen aus den Lernstandsanalysen Informatik kann folgendes Vorgehen als verallgemeinerbar auch für ähnliche Vorhaben in anderen Fächern betrachtet werden.

Rahmenbedingungen sichern

Untersuchungen zu Kompetenzen mit Schülern bedürfen zu Recht einer Begründung und der Genehmigung durch die Schulaufsicht. Dabei ist gerade durch eine Vielzahl von Studien (nicht nur durch PISA) teilweise eine Testhäufung vorhanden, die durch Hinarbeiten auf Tests einen systematischen Lernprozess im Zuge eines lehrplanorientierten Unterrichts beeinflussen. Im Fall der Lernstandsanalyse im Fach Informatik waren neben der Existenz eines Pflichtfaches mit einheitlichen Lehrplänen für die jeweilige Schulart auch die in großer Anzahl ausgebildeten Fachlehrer entscheidend für die Genehmigung einer breiten Untersuchung.

Aufgaben entwickeln

Für die Konstruktion von Aufgaben einer Lernstandsanalyse wurden zunächst zwei verschiedene Prozesse bewerkstelligt. Einerseits erfolgte eine detaillierte Analyse des Lernbereichs „Datenbanken“. Daraus resultierten eine vollständige Visualisierung des Themas bzw. dessen Kernkonzepte (vgl. Bo11). Andererseits wurden Experten benötigt (Lehrer aus der jeweiligen Schulart, Fachwissenschaftler und Kompetenzdiagnostiker – vgl. Vorwort), die nach einer Schulung zur Aufgabenkonstruktion aus psychologischer Sicht die Aufgaben ausarbeiten sollten. Im Mittelpunkt standen dabei Aspekte der Aufgabenoperatoren und Inhaltskomplexität, um ein erschöpfendes Bild über Kompetenzen der Schüler im ausgewählten Themengebiet zu erhalten. Auf der Basis dieser fachlichen Differenzierung der Teilgebiete wurden Expertengruppen aus den Lehrern und Wissenschaftlern gebildet, die Vorüberlegungen zu einzelnen Aufgaben trafen, diese ausarbeiteten und nach eingehender Diskussion auch modifizierten. Es war dabei das Ziel, nicht nur die inhaltlichen Schwerpunkte, sondern auch die kognitiven Operatoren [vgl. AK01] von einzelnen Fakten bis zu komplexen Gesetzmäßigkeiten [Me87] im Test zu variieren. Durch die Einordnung in eine Tylermatrix [Ni01], die durch den fachlichen Bereich als eine dritte Dimension ergänzt wurde, konnte dieser Prozess visualisiert (vgl. Abb. 1) und zielführend gestaltet werden, um zu vergleichbaren Aufgabensätzen für einen Vortest zu gelangen. Das Ergebnis der unterschiedlichen Entwicklungsetappen zeigt folgende Übersicht:

		Kognitive Operatoren					
		Erinnern	Verstehen	Anwenden	Analysieren	Bewerten	Erschaffen
inhaltliche Komplexität	Fakten	A 1 / V I	A 1 / V II			A 4 / V I A 3 / V II	
	Konzepte		A 6 / V I		A 2 / V I A 2 / V II		A 7 / V I A 7 / V II
	Prozeduren		A 4 / V II	A 3 / V I A 6 / V II			
	Gesetze			A 5 / V I A 5 / V II			

Abb. 1: Aufgabenstruktur mit allen Dimensionen für den Vortest

Die Aufgaben, die vor dem ersten Diskussions- und Modifikationsprozess durch die Experten erarbeitet wurden, hatten einen Schwerpunkt in den Operatoren Anwenden und Analysieren sowie in der inhaltlichen Komplexitätsstufe der Prozeduren. Da jedoch eine deutliche Reduktion der Aufgaben vorgenommen werden sollte (auf ca. 6 - 7 Aufgaben

passend zu einer 45-minütigen Testeinheit), hätte sich der Test möglicherweise nur auf diese beiden Anforderungen an die Schüler reduziert. In der Folge würden zum Beispiel Schüler, die das Thema Datenbanken zwar verstanden haben und sich an wichtige Fachbegriffe erinnern können, aber das Gelernte nicht zur Anwendung bringen können, keine oder nur sehr wenige Punkte erhalten. Es war notwendig, Aufgaben mit einer größeren Breite der kognitiven Anforderungen zu erarbeiten. Letztendlich konnte ein Aufgabenpool erstellt werden, der ausgewogen die inhaltlichen Schwerpunkte, die kognitiven Operatoren und die inhaltlichen Komplexitäten variiert, um möglichst viele Kompetenzen der Schüler in diesem Themengebiet zu diagnostizieren (vgl. Abb. 2).

Aufgabensätze testen

Der Vortest war auf etwa 5 % der erwarteten Teilnehmer (ca. 150 Schüler) ausgelegt und fand mit zwei Aufgabensätzen statt. Er wurde mit Schülern gleicher Lernerfahrungen aus Mittelschulen und Gymnasien durchgeführt und nutzte beide Varianten der Aufgaben. Damit konnten wichtige Voraussetzungen zur Beurteilung der Aufgabenschwierigkeit² und deren Trennschärfe³ geleistet werden. Die Schulnote in Informatik wurde als Maß genutzt, um in die Vortestgruppen ungefähr gleichviel gute oder weniger gute Schüler einzuordnen und so die Qualität zu sichern.

Variante 1	A 1	A 2	A 3	A 4	A 5	A 6	A 7
Variante 2	A 1	A 2	A 3	A 4	A 5	A 6	A 7

Abb. 2: Aufgabenschwierigkeit Vortest

Variante 1	A 1	A 2	A 3	A 4	A 5	A 6	A 7
Variante 2	A 1	A 2	A 3	A 4	A 5	A 6	A 7

Abb. 3: Trennschärfe Vortest

Sowohl die Aufgabenschwierigkeit als auch die Trennschärfe müssen bei einer Lernstandsanalyse bestimmten testtheoretischen Mindestwerten entsprechen. Beispielsweise sollte die Lösungswahrscheinlichkeit bzw. Schwierigkeit der Aufgaben zwischen 20 % und 80 % liegen [vgl. LR98]. Weiß hinterlegte Kästen bedeuten hier, dass diese Mindestwerte durch die Aufgabe im Vortest erfüllt werden konnten (leichte Grauwerte zeigen Aufgaben, die bereits leicht kritische Werte hatten; stärkere Grautöne zeigen Aufgaben, die diese Mindestwerte bereits leicht unterschritten haben). Deshalb wurden für den endgültigen Test die Aufgaben herausgesucht, die die besseren Werte hatten (bspw. A2 aus Variante 2; A5 aus Variante 1). Außerdem wurden A3 und A4 der ersten Variante erneut modifiziert und mit zusätzlicher Hilfestellung für eine

² Aufgabenschwierigkeit ist als die durchschnittliche Punktzahl definiert, die bei einer Aufgabe abhängig von der maximal möglichen Punktzahl durch alle Schüler erreicht werden konnte. Hohe Werte stehen dabei für (zu) leichte und niedrige Werte für (zu) schwere Aufgaben. Mittlere aber auch gut verteilte Werte bei der Aufgabenschwierigkeit sind Voraussetzung für weitere Qualitätskriterien (z.B. Trennschärfe) eines Tests.

³ Trennschärfe gibt an, ob eine Aufgabe präzise zwischen guten und schlechten Schülern unterscheiden kann. Sie wird als Korrelation zwischen der Lösung einer einzelnen Aufgabe und der in dem gesamten Test erzielten Punktzahl ohne diese Aufgabe berechnet.

erfolgreiche Bearbeitung versehen. Die Aufgabe A6 wurde mit Absicht nicht verändert und als sogenannte Eisbrecherfrage an den Anfang des Tests gestellt [vgl. ebenfalls LR98]. Diese Aufgabe war deutlich leichter, als die restlichen Aufgaben und sollte den Schülern im Test helfen, Vertrauen in ihre eigenen Fähigkeiten zu fassen.

Test durchführen

Die Durchführung eines solchen Tests mit Lehrern auf freiwilliger Basis erfordert neben der Motivation zur Teilnahme vor allem eine detaillierte und den Aufwand der Beteiligten bedenkende Planung. Über einen gesicherten WEB-Zugang erfolgte eine Erklärung zur Teilnahme [vgl. LS12] sowie die Bereitstellung des Erwartungsbildes und der ausgewerteten Ergebnisse. Der Aufgabensatz selbst wurde über die Schulleitungen versandt und am Tag der Durchführung dem jeweiligen Kollegen ausgehändigt. Für die Korrektur stand ein Erwartungsbild mit Itemcharakteristik zur Verfügung, auf dessen Basis die Auswertung der Arbeiten durch die Lehrer erfolgte und deren Ergebnisse (auch auf Itemebene) anonymisiert im WEB-Formular eingetragen wurden.

Ergebnisse auswerten

Nach Abschluss der Eintragung aller Ergebnisse wurde eine Gesamtauswertung durchgeführt und diese für beteiligte Schulen im Vergleich zu den eigenen Resultaten veröffentlicht. Außerdem standen die eingetragenen Ergebnisse (auf Itemebene) für jede Schule zur Nutzung bereit [vgl. LS12]. Anschließend begann eine differenziertere Auswertung der Resultate in der Expertengruppe.

3. Veranschaulichung der Aufgabenentwicklung am Beispiel

Anhand der fachlichen Bereiche des Themengebietes erstellten alle Experten erste Aufgabenvorschläge, die einen inhaltlichen Teilbereich abdecken sollten. Hier zeigte sich sofort die Schwierigkeit, diese aus fachlicher und didaktischer Sicht sinnvolle Struktur in geeigneten Beispielaufgaben abzubilden. An einem ausgewählten Beispiel soll dargestellt werden, wie sich die Erarbeitung der Aufgaben konkret vollzogen hat.

Die folgenden Daten zur Belegung von Kursen im GTA sollen in einer Datenbank abgebildet werden. Planen Sie eine geeignete Datenbasis! Stellen Sie die dazu notwendigen Tabellen und Beziehungen dar.									
SNr	Vorname	Name	Klasse	GTA-Kurs	KursNr	Tag	Zeit	Raum	Kursleiter
1123	Hermine	Hampel	5/3	10-Finger-Schreiben	001	Mittwoch	15:00	036	Frau Flott
1107	Gustav	Gärtner	6/2	10-Finger-Schreiben	001	Mittwoch	15:00	036	Frau Flott

Abb. 4: Beispielaufgabe aus dem Aufgabenpool

Diese erste Aufgabenidee (vgl. Abb. 4) wurde in der Expertengruppe vorgestellt und dabei war erkennbar, dass die Aufgabe aus fachlicher Sicht zu vielschichtig ist. Insbesondere musste die Formulierung unter Verwendung geeigneter Operatoren präzisiert werden. In weiteren Diskussionen wurden die vorgestellten Aufgabenideen nun nach mehreren Kriterien, wie fachlicher Bereich, kognitive Operatoren, inhaltliche Komplexität sortiert (vgl. Kap. 2) und zur Einordnung in die dargestellte Tylermatrix nach einer festen Vorgabe beschrieben (vgl. Abb. 5). So konnte erreicht werden, dass in einigen Bereichen der Matrix fehlende Aufgaben durch Modifikationen oder auch Neugestaltung ergänzt werden konnten.

Inhaltliche Ebene: Konzepte		Kognitive Anforderung: Analysieren	
Fachlicher Bereich: Modellierung / Implementierung			
Bearbeitungszeit: 10'	Gesamtpunkte: 8	Aufgabennummer: B1a	
Operator: analysieren			

Zielstellung
Die Schüler sind in der Lage, aus einem gegebenen Sachverhalt ein geeignetes Datenmodell zu erstellen.

Begründung der verwendeten Operatoren
Eine Menge gegebener Informationen müssen in Klassen und Attribute zerlegt werden. Anschließend werden die Attribute den entsprechenden Klassen zugeordnet und in das Schema übertragen.
Die mengenmäßige Zuordnung muss aus zwei Blickrichtungen (Hin- und Rückrichtung) unterschieden werden.

Abb. 5: Einordnung der Aufgabe

Nach weiteren Diskussionen in der Expertengruppe wurde im Vortest folgende Aufgabe in zwei Varianten genutzt (Abb. 6 und 7):

In einem Zoo leben die *Tiere* in verschiedenen *Gehegen*. Zur Organisation der Pflege und des Fütterns, werden die folgenden Informationen zum Gehege und den Tieren in einer Datenbasis erfasst:
Herkunft, Gehegeart, Geburtsdatum, Größe des Geheges, Futter, Tiername, Masse

Aufgabenstellung
a) Analysieren Sie die Informationen und ergänzen Sie das Datenmodell in dem gegebenen Schema.

The diagram shows an Entity-Relationship (ER) model. At the top, there are two entity sets represented by rectangles. The left entity has two attributes (ovals) below it. The right entity has four attributes below it. A diamond-shaped relationship set connects the two entities. Below this, there are two empty rectangular boxes, one on the left and one on the right, with arrows pointing towards each other, indicating where the student should complete the model.

Abb. 6: Vortestaufgabe 1, Variante I

In der anderen Variante dieser Vortestaufgabe wurde nur der Einstiegstext geändert.

Jeder *Handyhersteller* entwickelt und vertreibt verschiedene *Modelle*. Um den Verkauf in den Geschäften zu erleichtern, werden die folgenden Informationen in einer Datenbasis erfasst:
 Displaygröße, Firmensitz, Herstellername, Kamera, Listenpreis, Modellname, Simkartengröße

Abb. 7: Vortestaufgabe 1, Variante II

Um dem Schüler eine Hilfestellung beim Modellieren zu geben und die Korrektur eindeutig zu gestalten, wurden sowohl das ER-Modell als auch das Klassenmodell als Lösungsansatz vorgegeben. Anhand der Auswertung des Vortestes diskutierten die Experten nochmals über die Operatoren und einigten sich auf folgende Formulierung der Aufgabenstellung für die finale Version der Testaufgabe (Abb. 8):

Abb. 8: Testaufgabe – finale Version

In der Korrekturanleitung erhielten alle beteiligten Kollegen mit dem Erwartungsbild eine detaillierte Auflistung der einzelnen Punkte. Diese Punktverteilung fand sich im webbasierten Auswertungstool [LS12] wieder. Es wurde in jeder Aufgabe (Abb. 9) pro Schüler erfasst, ob der einzelne Punkt gegeben oder die Aufgabe nicht gelöst wurde.

Aufgabe 6		P:	N:
a)	sinnvolle Benennung der Klasse HERSTELLER	<input type="checkbox"/>	<input type="checkbox"/>
	sinnvolle Benennung der Klasse MODELL	<input type="checkbox"/>	<input type="checkbox"/>
	Benennung der Beziehung	<input type="checkbox"/>	<input type="checkbox"/>
	Zuordnung aller Attribute (HERSTELLER)	<input type="checkbox"/>	<input type="checkbox"/>
	Zuordnung aller Attribute (MODELL)	<input type="checkbox"/>	<input type="checkbox"/>

Legende:
 P=Frage korrekt beantwortet
 N=Frage nicht beantwortet

bisher eingegebene Daten:
 Vom Nutzer "Rätz" wurden bisher 1 Schüler gespeichert.

ID des **zuletzt** gespeicherten Schülers:
 829 (30 Punkte)

(ID bitte notieren, falls Sie die gespeicherten Ergebnisse nachträglich Ihren Schülern zuordnen möchten)

Abb. 9: Ausschnitt aus dem Auswertungstool

4. Ergebnisse und Perspektiven

Bei der folgenden Darstellung der Resultate wird zwischen Ergebnissen zur Stichprobe und Testqualität sowie Ergebnissen zu Aufgaben und Items unterschieden.

Stichprobe und Testqualität

In die Lernstandsanalyse konnten 2318 Schüler einbezogen werden, davon waren 957 weiblich und 1282 männlich (79 Einträge sind ohne eine Angabe in die Auswertung eingeflossen). 749 Schüler besuchten zum Erhebungszeitpunkt das Gymnasium und 1569 Schüler die Mittelschule. Außerdem stammten 54 Schüler aus dem Bundesland Thüringen. Die 2318 Schülerinnen und Schüler konnten insgesamt 68 Schulen zugeordnet werden, davon 63 Schulen aus Sachsen. Die Lernstandsanalyse basiert damit auf einer großen Stichprobe aus Schülerinnen und Schülern.

Mit der Aufgabenschwierigkeit und der Trennschärfe wurden bereits zwei der wichtigen Qualitätskriterien einer Lernstandsanalyse beschrieben (vgl. Kap. 2). Die dort gewählte Visualisierung der testtheoretischen Mindestwerte soll auch hier - für die letztlich verwendeten Aufgaben der Lernstandanalyse - Verwendung finden.

Test	A 1	A 2	A 3	A 4	A 5	A 6	A 7
------	-----	-----	-----	-----	-----	-----	-----

Abb. 10: Aufgabenschwierigkeit und Trennschärfe der Lernstandsanalyse

Es fällt dabei auf, dass die Aufgaben 2 - 7 die Mindestwerte vollständig erfüllten (weiße Hinterlegung). Die Aufgabe 1 (ehemalig A6 im Vortest) wurde als Eisbrecherfrage konstruiert und hat damit bewusst nicht die eigentlich notwendige Schwierigkeit bzw. Lösungswahrscheinlichkeit und somit auch keine hinreichende Trennschärfe. Die Modifikationen nach dem Vortest waren damit erfolgreich. Auch bei Betrachtung weiterer Kriterien kann die LSA aus testtheoretischer Sicht als korrekt bewertet werden.

Die Schülerinnen und Schüler konnten in der LSA maximal 42 Punkte erreichen. Davon erhielten sie durchschnittlich 24,85 Punkte. Die Standardabweichung betrug 8 Punkte. Die Verteilung des Gesamtergebnisses entspricht annähernd einer Normalverteilung (vgl. Abb. 11). Auch dies spricht für eine gute Testqualität.

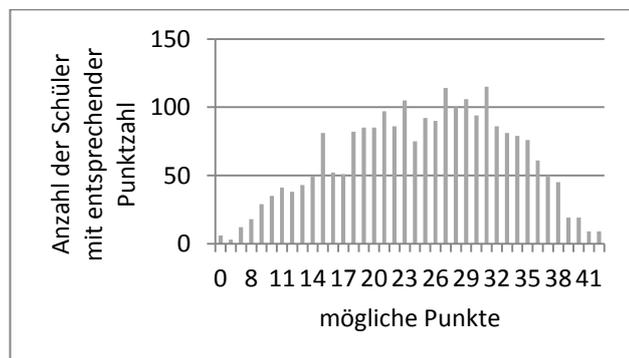


Abb. 11: Punkteverteilung

Im Allgemeinen wird häufig davon ausgegangen, dass in Schulfächern wie Informatik Jungen bessere Leistungen im Vergleich zu Mädchen erzielen. Deshalb wurde in der hier beschriebenen Lernstandsanalyse eine differenzierte Betrachtung des Zusammenhangs von Geschlecht und Gesamtpunktzahl vorgenommen. Im Mittel erreichten Mädchen 25,26 Punkte und Jungen 24,59 Punkte. Damit haben weibliche Teilnehmende sogar einen minimal größeren Wert, als ihre männlichen Mitschüler. Ein Test auf statistische Signifikanz ergab jedoch, dass dieser Unterschied nicht bedeutsam ist (einfacher T-Test; $t(2237) = 8.298, p > .05$). Folglich haben beide Geschlechter vergleichbare Resultate erzielt.

Erstmalig wurde der Versuch unternommen, auch ein anderes Bundesland mit in die Lernstandsanalyse zu integrieren. Da jedoch keine einheitlichen Lehrpläne für die verschiedenen Bundesländer existieren, kann hier zunächst nur von einem Testlauf gesprochen werden. 5 Schulen mit insgesamt 54 Schülern aus Thüringen beteiligten sich an der Lernstandsanalyse. Sie erreichten im Durchschnitt einen Wert von 28,28 Punkten (Standardabweichung = 11,18). Damit erzielten diese Schüler sehr gute Resultate. Auch wenn hier kein Test auf signifikante Unterschiede vorgenommen werden sollte, kann von einem sehr erfolgreichen Testlauf gesprochen werden.

Resultate auf Aufgaben- bzw. Itemniveau

In der folgenden Tabelle sind noch einmal alle Aufgaben der Lernstandsanalyse abgetragen. Zusätzlich sind neben einer inhaltlichen Einordnung die erreichten Punkte als Mittelwert, die Standardabweichung und die Lösungswahrscheinlichkeit angegeben.

Aufgabennummer	fachlicher Bereich	erreichbare Punkte	Mittelwert	Standardabweichung	Wahrscheinlichkeit
1	Eisbrecher-Aufgabe	7	6,32	1,071	90 %
2	Grundbegriffe	5	3,18	1,165	64 %
3	Datenbankabfragen prüfen	3	1,83	1,147	61 %
4	Datenbankabfrage planen	7	3,09	2,404	44 %
5	Erzeugung eines Relationenmodells	6	2,54	1,899	42 %
6	Erzeugung eines Datenmodells (mit Unterstützung)	8	4,90	2,510	61 %
7	Erzeugung eines Datenmodells (ohne Unterstützung)	6	3,00	1,948	50 %

Abb. 12: Übersicht über alle Aufgaben der Lernstandsanalyse

Die Daten in der Tabelle zeigen, dass die Aufgaben im Mittel sehr unterschiedlich, insgesamt gesehen gut, durch die Schüler bearbeitet wurden. So gibt es wichtige Kompetenzbereiche, bei welchen einige Schüler keine oder nur wenige Punkte erwerben konnten – andere Kompetenzbereiche beherrscht diese Zielgruppe demgegenüber eher

gut. Außerdem kann jede Aufgabe und damit jeder Kompetenzbereich noch in kleine, einzelne Kompetenzen zerlegt werden. So sind bei der Erzeugung eines Datenmodells beispielsweise verschiedenste Anforderungen an die Schüler vorhanden (z.B. Analyse der in der Aufgabe genannten Problemstellung, Erzeugung eines Datenmodells oder Bestimmung des Beziehungstyps). Auch hier wird demnächst eine tiefere Analyse auf Punkt- bzw. Itemebene (auch zwischen verschiedenen Aufgaben) erfolgen.

Perspektiven der Lernstandsanalyse

Die Ergebnisse der Lernstandsanalyse zeigen die Qualität des Tests, die durch das aufwendige Vorgehen bei der Aufgabenkonstruktion erzielt werden konnte. Jede Aufgabe trägt individuell zur Diagnose der Kompetenzen bzw. der Schülerleistung bei. Dadurch sind didaktische Konsequenzen strukturiert ableitbar. Dies wird zeitnah durch weitere Analyseschritte mit verschiedenen Fachvertretern erreicht. Erste Ergebnisse wurden bereits im Kapitel 4 beschrieben.

Zusammenfassend kann das geschilderte Vorgehen als Workflow für die Entwicklung einer Lernstandsanalyse betrachtet werden. Die dargestellte Lernstandsanalyse kann als Vorbild für andere Inhaltsbereiche, Fächer oder Standorte dienen und liefert Anregungen für weitere Untersuchungen. Beispielsweise wäre die Analyse der Ausgangsbedingungen zum Informatikunterricht am Anfang der Klassenstufe 7 oder die Kompetenzen am Ende der Grundschule ein möglicher Untersuchungsgegenstand für die informatische Bildung und den Informatikunterricht in Sachsen.

Literaturverzeichnis

- [BKF09] Katrin Büttner, Thomas Knapp, Steffen Friedrich: Untersuchungen zu informatischen Kompetenzen in Sachsen - eine empirische Studie. In (Koeber, B. Hrsg) Zukunft braucht Herkunft. 25 Jahre "INFOS – Informatik und Schule“ Berlin 2009; S. 62-72
- [BS07] Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik für die Sekundarstufe I. Gesellschaft für Informatik e.V., Bonn In: LOG IN 150/151 (2008)
- [LP04] Lehrplan Informatik für Mittelschulen und für Gymnasien unter:
<http://www.bildung.sachsen.de/apps/lehrplandb/> (08.02.2013)
- [Bo11] Bollwig, Ch., Lernstandsanalyse in Informatik – kompetenzorientierte Aufgabenentwicklung. Staatsexamensarbeit. TU Dresden, Juni 2011
- [AK01] Anderson, L. W., & Krathwohl, D. R. (Eds.). A taxonomy of learning, teaching, and assessing. New York: Longman, 2001.
- [Me87] Merrill, M.D. A lesson based upon Component Display Theory. In C. Reigeluth (ed.), Instructional Design Theories in Action. Hillsdale, NJ: Erlbaum Associates, 1987.
- [Ni01] Niegemann, H.M., Domagk, S., Hessel, S., Hein, A., Hupfer, M. & Zobel, A.: Kompendium multimediales Lernen. Heidelberg: Springer, 2001.
- [LS12] <http://www.sn.schule.de/~lsa-informatik>
- [LR98] Lienert, G., & Raatz, U. Testaufbau und Testanalyse. Weinheim: Psychologie Verlags Union, 1998.

Ein moderner Ansatz für Anchored Instruction im Informatikunterricht

Jens Gallenbacher, Dominik Heun

Technische Universität Darmstadt
Didaktik der Informatik
Hochschulstraße 10
64289 Darmstadt
jg@di.tu-darmstadt.de
dh@di.tu-darmstadt.de

Abstract: Die Informatik ist eine Wissenschaft, die einerseits durch Formalien aus der Mathematik und andererseits aus Ingenieurwissenschaften entstand. Dadurch verfügen viele Bereiche der Informatik über einen hohen Anwendungsbezug. Diesen gilt es motivierend in den Informatikunterricht zu integrieren.

Dieses Dokument stellt den Ansatz der *anchored instruction* vor, der schon 1990 entwickelt wurde. Unter Berücksichtigung bekannter Konzepte der Informatikdidaktik wird dieser Ansatz eingeordnet und weiterentwickelt. Mit dem außerschulischen Lernlabor *Abenteuer Technik* wird ein konkretes Beispiel für die Verwendung des eingeführten Ansatzes dargestellt.

1 Anchored Instruction

1.1 Begriffsklärung

Der Ansatz *anchored instruction* wurde 1990 an der Vanderbilt Universität in Nashville, Tennessee entwickelt und beschreibt eine Lernumgebung, die sich durch eine Rahmenschicht auszeichnet. Innerhalb dieser Geschichte werden die Lernenden mit Problemen konfrontiert, die sie selbstständig lösen sollen. Dazu erhalten sie sämtliche nötige Informationen aus der Geschichte, die in vielfältiger Weise präsentiert werden können.

Der Ansatz wurde entwickelt, um im Mathematikunterricht der Jahrgangsstufe 5 die Problemlösefertigkeit zu stärken. Des Weiteren sollte der Tendenz des *trägen Wissens*, des Wissens ohne anwendungs- oder realitätsbezogenem Kontext, entgegengewirkt werden.

Anfangs wurden fremdproduzierte Filme (bspw. Young Sherlock Holmes oder Jäger des verlorenen Schatzes) eingesetzt, um über mathematische Probleme ins Gespräch zu kommen. In späteren Fassungen wurden das Material selbstproduziert, woraus die Jasper-Stories entstanden. Innerhalb dieser Filmclips wird der Hauptdarsteller Jasper Woodbury in schwierige Situationen versetzt, die die Lernenden dann auflösen sollen. Durch die Eigenproduktion der Materialien konnten Inhalte abgedeckt werden, die durch kommerzielle Filme nicht erreicht wurden.

Mit dem Einsatz der neuen Methode sollte die Wissensvermittlung motivierender gestaltet werden sowie das erlernte Wissen nachhaltiger verfügbar sein.

1.2 Aufbau von Materialien

Materialien, die nach dem *anchored instruction*-Ansatz aufgebaut sind, können durch sieben Kriterien verdeutlicht werden¹.

- Videounterstützung
- Reale Anwendungsbereiche
- Generativ
- Daten in Geschichte eingebettet
- Komplexität des Problems
- Vergleichbare Stories
- Aufzeigen inter- und intradisziplinärer Bezüge

Die Videounterstützung wird gewählt, um einerseits die Lernenden zu motivieren und andererseits leseschwachen Schülerinnen und Schülern entgegenzukommen. Des Weiteren können Video erneut angeschaut und so während des Problemlöseprozesses mehrfach zu Rate gezogen werden.

Durch realistische Probleme soll dem *trägen Wissen* entgegengewirkt werden. In der Mathematik, wo *anchored instruction* seinen Ursprung besitzt, war das abstrakte Rechnen weit verbreitet. Durch Alltagsbezüge sollen die Lernenden motiviert werden, indem aufgezeigt wird, wo Mathematik verwendet wird. Ob dieses Argument noch relevant ist, kann jeder Lehrende beantworten, dessen Lerngruppe sagt: „Das brauche ich später nie wieder!“. Des Weiteren soll eine flexiblere Nutzung des Wissens angestrebt werden. Nur die Schülerinnen und Schüler, die schon einmal reale Probleme mit Mathematik gelöst haben, werden Probleme in ihrem Alltag mit den Strategien angehen können. Lernende, die nicht in diesem Maße geschult wurden, müssen in Problemsituationen den Transfer erstmals erbringen, was einen Misserfolg begünstigt.

Der generative Aufbau der Probleme soll die Schülerinnen und Schüler motivieren das Ende der Geschichte zu finden. Mit dem Ende des Videos sind die zu lösenden Probleme nicht klar definiert. An dieser Stelle sind die Lernenden aufgefordert die Probleme zu identifizieren und einen Plan zu entwerfen, mit dem eine Lösung erreicht werden kann.

Alle Daten, die zur Lösung des Problems notwendig sind, können in der Geschichte gefunden werden. Dabei ist eine zusätzliche Aufgabe relevante von irrelevanten Daten zu unterscheiden. Dies wirkt erneut dem *trägen Wissen* entgegen, da sich die Lernenden konkrete Gedanken über das Problem machen müssen. Bekannt sind in diesem Kontext aus

¹vgl. [Va97, S. 46 ff]

der Mathematik sogenannte Kapitänsaufgaben wie bspw. „Auf einem Schiff befinden sich 14 Schafe und drei Hühner. Wie alt ist der Kapitän?“. Mehr als die Hälfte der Lernenden der 5. - 7. Jahrgangsstufe führen bekannte Rechenoperationen aus, in denen sie Angaben verwenden, ohne dabei deren Bedeutung zu hinterfragen.²

Die Motivation der Teilnehmenden das Problem zu lösen, wird zusätzlich verstärkt, da das Ende der Geschichte nur dann erzählt werden kann.

Ein Problem nach dem anchored instruction Ansatz sollte nicht zu einfach zu lösen sein. Damit wird das einfache Ausprobieren von unterschiedlichen Ansätzen unterbunden. Die Lernenden müssen vielmehr mit einem systematisch vorgehen.

Stoßen die Lernenden auf vergleichbare Aufgaben, werden sie unter Umständen versuchen eine bekannte Lösung zu nutzen. Dabei lernen sie, wann eine Problemlösung übertragen werden kann und unter welchen Umständen eine Analogiebildung ausgeschlossen ist.

Mit dem Aufzeigen von Bezügen zur eigenen Fachwissenschaft wird die Strukturierung der Disziplin verdeutlicht. Verbindungen zu anderen wissenschaftlichen Disziplinen können helfen, die vielfältige Verwendung eines Faches aufzuzeigen.

1.3 Lernförderlichkeit

Nach der Entwicklung der anchored instruction durch die Vanderbilt Universität 1990, forschten unterschiedliche Institute, ob anchored instructions lernförderlich sind. Dabei wurde aus den jeweiligen Erkenntnissen der Forschungsgegenstand weiterentwickelt, um ein bestmögliches Ergebnis zu erzielen.

So führte Pellegrino 1991 eine Studie zu ursprünglichen Jasper-Stories durch und konnte aufzeigen, dass die Experimentalgruppe mit anchored instructions zu besseren Ergebnissen in Mathematik kam, als die Kontrollgruppe. Dabei stellte die Forschungsgruppe aber auch fest, dass die Streuung der Ergebnisse sehr hoch war³.

Griffin untersuchte 1995 den Unterschied im Lernerfolg von anchored instruction im Vergleich zu herkömmlichem Unterricht. Zu diesem Zeitpunkt herrschte hier der Instruktivismus vor. In einem Handlungstest (Lösung einer Aufgabe) konnten signifikante Verbesserungen festgestellt werden. Eine Transferaufgabe wurde aber nicht signifikant besser bearbeitet⁴.

Insgesamt führt [Blu03] noch weitere Studien auf, die einen positiven Einfluss von anchored instruction auf das Lernverhalten vermuten lassen. Hickey et. al zeigte allerdings 2001 auf, dass bestimmte Fähigkeiten der Schülerinnen und Schüler durch Anwendung der Jasper-Stories verringert werden. So konnte nachgewiesen werden, dass die Lernenden der Testgruppe klassische Rechenaufgaben nicht so effizient gelöst haben, wie die Kontrollgruppe. Die Problemlösefähigkeit sowie die Interpretationsfähigkeit von Daten konnte allerdings signifikant verbessert werden.

²vgl. [Bar89, Seite]

³vgl. [Blu03, S. 41 f]

⁴vgl. [Blu03, S.50 f]

Zusammenfassend zeigt sich in den Studien kein eindeutiges Bild zur Lernförderlichkeit der anchored instruction Methode. In vielen Bereichen konnten Verbesserungen gegenüber traditionellen Unterrichtsmethoden festgestellt werden, die aber auch zu einer höheren Streuung führten. Aspekte, die durch diese Methode gefördert werden sollen, verbessern sich. Im Gegenzug werden aber traditionelle Fähigkeiten wie Rechenfähigkeiten vernachlässigt.

2 Notwendigkeit von anchored instructions?

Die Entwicklung des *anchored instruction*-Ansatzes wurde aus der Situation der Mathematik im Jahr 1990 heraus betrieben. In diesem Abschnitt soll erörtert werden, inwiefern die Situation, die 1990 zur Entwicklung der anchored instruction im Mathematikunterricht geführt hat, auf den Informatikunterricht von heute übertragbar ist.

2.1 Die Situation der Mathematik

1990 herrschte im Mathematikunterricht keine mit heute vergleichbare Aufgabenkultur. Das sogenannte *Päckchenrechnen* war weit verbreitet und somit verantwortlich für die Produktion trägen Wissens. Des Weiteren waren die Auswirkungen der sogenannten *Neuen Mathematik-Bewegung* zu spüren, die eine Ausrichtung an der Fachwissenschaft Mathematik bewirkte. Doch seitdem hat sich die Mathematikdidaktik weiterentwickelt und versucht diesem Trend entgegenzuwirken.

Im heutigen Mathematikunterricht ist ein vielfältiger Umgang mit Aufgaben vorgesehen. So werden bspw. Aufgaben über Gegebenem, Gesuchtem und Rechenweg analysiert⁵ und in einem abwechslungsreichen Maße im Unterricht eingesetzt. Aufgaben werden geöffnet, damit sie die Lernenden kognitiv auf mehreren Ebenen fordern und fördern. Anwendungsorientierte Aufgaben zeigen im Mathematikunterricht reale Bezüge zur Außenwelt auf.

Der Mathematikunterricht hat sich verglichen zu 1990 enorm gewandelt und seine Aufgabenkultur vielförmig weiterentwickelt. Für den Informatikunterricht steht die Entwicklung einer Aufgabenkultur noch weit am Anfang, weshalb bisher auf keine Erkenntnisse für den Informatikunterricht zurückgegriffen werden kann. Mit Anwendungsbezügen hat sich die Informatikdidaktik ausführlich beschäftigt, wie im nächsten Abschnitt dargelegt werden soll.

2.2 Entwicklung innerhalb der Informatik-Didaktik

Die Entwicklung der Informatik kann aus zwei Richtungen betrachtet werden. Zum einen hat die Informatik einen formalen Kern, den sie aus der Mathematik geerbt hat. Zum an-

⁵siehe dazu [BBKL08]

deren weist sie aber auch einen hohen Anteil einer Ingenieursdisziplin auf, durch den sie vielfältige Kontexte umfasst, die in der Schule durch Anwendungen und Projektarbeiten realisiert werden können.

Ansätze aus der Informatik-Didaktik hin zu einem anwendungsorientierten Informatikunterricht finden sich bspw. in [SZ78], wo eine *Erziehung zur Handlungsfähigkeit* gefordert wird. Die Betroffenheit der Lernenden mit einem Lerngegenstand ist zentraler Aspekt des anwendungsorientierten Informatikunterrichts. Die Betrachtung von fertigen Computerprogrammen anstatt der Programmierung innerhalb der informationstechnischen Grundbildung, ließ den Begriff der *Anwendungsorientierung* aber jede Trennschärfe, und damit Bedeutung für eine Steigerung der Lernförderlichkeit im Informatikunterricht verlieren⁶.

*Informatik im Kontext*⁷ ist eine Entwicklung innerhalb der Informatik-Didaktik, die in Bezugnahme auf die kontextorientierten Initiativen der mathematisch-naturwissenschaftlichen Fachdidaktiken (bspw. Chemie im Kontext), die folgenden Kriterien für kontextorientierten Informatikunterricht entwickelt hat:

- Orientierung an Kontexten
- Orientierung an Standards für die Informatik in der Schule
- Methodenvielfalt

Grundelement der anchored instruction, die Orientierung an realen Kontexten, findet sich hier wieder. Kontexte im Sinne von *Informatik im Kontext* müssen aber neben einem realen Bezug auch Bezüge zum „Erfahrungshorizont der Schülerinnen und Schüler“ aufweisen⁸. Des Weiteren werden die Bildungsstandards der Gesellschaft für Informatik⁹ als zu erreichende Ziele definiert.

2.3 Nachteile von anchored instruction

Lernmaterialien zu den Kriterien der anchored instruction herzustellen ist sehr aufwendig, da geeignetes Filmmaterial produziert werden muss. Dies ist für den Unterricht in der Schule nicht zu leisten. Des Weiteren gibt es keinen Fundus für bereits produzierte Filme, aus dem Materialien für den Unterricht entnommen werden können.

In anchored instruction-Medien werden Schülerinnen und Schüler mit realen Anwendungen konfrontiert. Sie erleben die Probleme aber nicht selbst, sondern sind lediglich Beobachter. Somit müssen sie die Videosequenz verarbeiten und das Problem erkennen. Dies ist mit realen Problemen nicht vergleichbar, da dabei Probleme selbst erfahren werden. Lernende, die diesen Transfer nicht leisten können, können auch nicht von anchored-instruction profitieren.

⁶vgl. [KSSW09, S. 4]

⁷siehe dazu [KSSW09]

⁸vgl. [KSSW09]

⁹[GfI07]

Anchored instruction setzt konstruktivistisches Lernen voraus. Informationsbeschaffung und fachliche Inhalte sind so nur schwer möglich. Mithilfe des gemäßigten Konstruktivismus¹⁰, der auf eine Kombination aus Instruktionen und konstruktivem Lernen setzt, können diese Probleme angegangen werden.

Wie in den vorgestellten Studien zur Lernförderlichkeit sichtbar wird, eignen sich anchored instructions nicht für alle Bereiche des Lernens. Transferleistungen können durch anchored instruction verbessert werden, wohingegen das Aneignen von neuem Wissen durch anchored instruction nur motiviert werden kann.

3 Anchored instruction im Lernlabor *Abenteuer Technik*

Mit dem Lernlabor *Abenteuer Technik* hat der Fachbereich Informatik der Technischen Universität Darmstadt einen außerschulischen Lernort für MINT-Fächer (Mathematik, Informatik, Naturwissenschaften und Technik) etabliert. Seit Mai 2011 werden regelmäßig Workshops für Schulklassen angeboten.

Die konzeptionelle Ausrichtung des Lernlabors beruht auf den folgenden Prinzipien:

- Spielerisches Lernen
- Genetischer Aufbau der Lernumgebung
- Kompetenzorientierung anhand der Bildungsstandards der Gesellschaft für Informatik, basierend auf den beiden informatischen Grundkompetenzen Modellbildung und Problemlösen
- anchored instruction
- low-tech

Der folgende Abschnitt soll die Umsetzung der anchored instruction im Lernlabor verdeutlichen und Veränderungen gegenüber der ursprünglichen Definition deutlich machen. Abschließend werden die neuen Kriterien anhand eines Beispiels dargestellt.

Martin Wagenschein bemerkte 1999 [WBH10] "Verstehen heißt: Selber einsehen, 'wie es kommt'." Das direkte Erleben eines Experiments und dessen Lerneffekt kann niemals durch eine multimediale Präsentation ersetzt werden und wird nur durch das eigene Erfahren übertroffen. Aus diesem Grund ist reales Experimentieren im Unterricht von entscheidender Bedeutung.

Daher verfolgt Abenteuer Technik einen neuen Ansatz für anchored instruction, der auf positiven Vorerfahrungen, aber auch auf den oben erwähnten Erkenntnissen beruht. Weg vom abstrakten digitalen Medium, hin zum Rollenspiel mit realen Objekten als Mittel zur Problemlösung, das durch entsprechende Effekte wie Projektion, Licht- und Toneffekte

¹⁰siehe [SM02, S. 149 f]

atmosphärisch unterstützt wird. Ein Beispiel dafür kann man im erfolgreichen Science-Center Odysseum in Köln sehen, wo bewusst Cinematographie eingesetzt wird, um junge Forscherinnen und Forscher aufnahmefähiger zu machen.

Durch Filmmaterial werden weder die zu lösenden Probleme, noch aufgabenrelevante Informationen weitergegeben. Der Film stellt lediglich den Anker dar und versetzt die Lernenden in eine spezifische Situation, die im weiteren Verlauf durch das Rollenspiel der Betreuenden vertieft wird. Aufgabenstellungen werden durch einen handlungsorientierten Ansatz innerhalb des Rollenspiels vermittelt und erlauben so eine direktere Erfahrung durch die Teilnehmenden.

Kontexte mit denen Lernende bei Abenteuer Technik konfrontiert werden, müssen aber nicht wie bspw. in *Informatik im Kontext* aus der unmittelbaren Lebenswelt der Lernenden stammen. Das Rollenspiel erlaubt die Einbettung von nicht-alltäglichen Beispielen. Die verwendeten Bezüge sind dennoch realistisch und zeigen auf, dass Ideen der Informatik in sehr vielen Bereichen verwendet werden können. Dies unterstützt einerseits den interdisziplinären Gedanken der anchored instruction und andererseits unterstreicht es den Allgemeinbildungscharakter der Informatik¹¹.

Dieser veränderte Ansatz ist durchaus als Unterrichtsmethode auch direkt in den Schulen möglich. Einige Lehrerinnen und Lehrer adaptierten nach einer Teilnahme an einem der Workshops das Konzept bereits. Das notwendige Filmmaterial stellen wir dazu etwa über Plattformen wie bspw. YouTube zur Verfügung.

4 Beispiel: Workshop *Jubel, Trubel, Informatik*

Dieser Abschnitt stellt ein konkretes Beispiel für die geänderte Verwendung des anchored instruction-Begriffs dar. Dabei handelt es sich um einen Workshop für Schulklassen der Jahrgangsstufe 7 und 8, in dem innerhalb eines Schultages Konzepte der Graphentheorie und allgemeine Problemlösestrategien vermittelt werden.

4.1 Hintergrundgeschichte

Im vorangegangenen Jahr fand in Darmstadt ein Jahrmarkt statt, der von der Stadt Darmstadt ausgerichtet und organisiert wurde. Ein Jurist, ein Schatzmeister und ein Magier waren jeweils für bestimmte Bereiche der Planung verantwortlich. So sorgte der Jurist für die Einhaltung bestehender Gesetze und setzt Verträge mit potentiellen Schaustellern auf. Die Aufgaben des Schatzmeisters drehen sich um die Einsparung von Geld durch Optimierung bestehender Abläufe. Für die Unterhaltung ist der Magier zuständig, der Zaubertricks von anderen Magiern kopieren möchte. Durch mangelhafte Planung und Kommunikation wurde der Jahrmarkt ein großer Reinflall.

In diesem Jahr erhalten die selben Personen ein zweite Chance und werden dabei von den

¹¹vgl. [Wit03]

Generell sind die Aufgaben so konzipiert, dass eine schrittweise konstruierende Lösung durch die Teilnehmenden möglich ist. Hilfestellungen erhalten sie durch den Organisator nur bedingt, da dieser in seiner Rolle als Jurist nicht mit Prinzipien der Informatik vertraut ist. Deshalb sind innerhalb des Lernlabors verschiedene Zeitschriften ausgelegt, die mögliche Lösungsansätze vorstellen. Dabei werden die Lösungshinweise schülergerecht aufbereitet in bspw. einem Interview innerhalb der Zeitschrift. Zusätzlich dienen die Zeitschriften dazu, einen Alltagsbezug der verwendeten informatischen Strategien vorzustellen. Dieser kann z.B. in Form einer Reportage präsentiert werden.

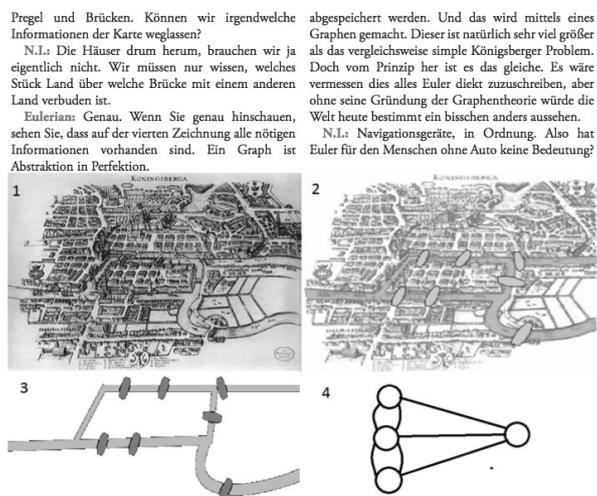


Abbildung 2: Ausschnitt einer Zeitschrift mit Hilfestellungen

Unter Nutzung der Zeitschriften wird die reale Wissensbeschaffung aus Büchern oder Zeitschriften simuliert, die die Lernenden im weiteren Verlauf ihrer Schulbildung anwenden müssen. Ferner bieten die Zeitschriften durch ihre abwechslungsreiche Gestaltung einen weiteren Motivationsfaktor. Ein Beispiel für die Motivation eines Graphen durch ein Interview ist in Abbildung 2 zu sehen.

Haben die Teilnehmenden die Planung erfolgreich abgeschlossen, werden sie mit einem Abschlussfilm, der das erfolgreiche Stadtfest zeigt, verabschiedet.

5 Zusammenfassung und Ausblick

Anchored instruction und *Informatik im Kontext* bieten vielversprechende Ansätze zur Kontextorientierung des Informatikunterrichts. Das Lernlabor *Abenteuer Technik* verwendet Intentionen beider Richtungen und entwickelt diese weiter zu einem umfassenden Rollenspielerlebnis, bei dem Teilnehmende des Workshops in die Rolle von Experten verschiedener Aufgaben gesetzt werden. Aufgabenstellungen unterschiedlichster Art unter-

streichen die Bedeutung der Informatik im allgemeinbildenden Unterricht und flexibilisieren die Anwendung der erlernten Kompetenzen.

In über 70 Workshops seit Mai 2011 konnte dieses Konzept erfolgreich mit etwa 2000 Schülerinnen und Schülern umgesetzt werden. Etliche Rückmeldungen von Lehrerinnen und Lehrern, die mit neuen Klassen den außerschulischen Lernort besuchen, bestätigen den Ansatz.

Der Bezug zu den präsentierten Geschichten wird demnach immer noch zitiert, die erworbenen Kompetenzen dadurch nachhaltig abrufbar. Explizit wurde einige Male auf den Unterschied zum Besuch eines Science-Centers hingewiesen, dessen Effekt nach wenigen Tagen verfliegen sei.

Ein Ziel der nächsten Jahre ist, diese *Anekdotische Evidenz* durch empirische Erhebungen zu belegen.

Literatur

- [Arl78] Wolfgang Arlt. *EDV-Einsatz in Schule und Ausbildung*. Oldenbourg, München, 1978.
- [Bar89] S. Baruk. *Wie alt ist der Kapitän? - über den Irrtum in der Mathematik*. Birkhäuser, Basel, 1. Aufl., Auflage, 1989.
- [BBKL08] Regina Bruder, Andreas Büchter, Evelyn Komorek und Timo Leuders. *Mathematikunterricht entwickeln - Bausteine für kompetenzorientiertes Unterrichten*. Cornelsen Vlg Scriptor, Berlin, 2008.
- [Blu03] P. Blumschein. *Eine Metaanalyse zur Effektivität multimedialen Lernens am Beispiel der Anchored Instruction*. Dissertation, Albert-Ludwigs Universität Freiburg, 2003.
- [Gfi07] Gesellschaft für Informatik. Grundsätze und Standards für die Informatik in der Schule. *LOG IN*, 146/147, 2007.
- [KSSW09] Jochen Koubek, Carsten Schulte, Peter Schulze und Helmut Witten. Informatik im Kontext (IniK) - Ein integratives Unterrichtskonzept für den Informatikunterricht. In *INFOS*, 2009.
- [SM02] Gerald A. Straka und Gerd Macke. *Lern-Lehr-Theoretische Didaktik* -. Waxmann Verlag, Münster, 2002.
- [SZ78] Renate Schulz-Zander. Analyse curricularer Ansätze für das Schulfach Informatik. In [Arl78], 1978.
- [Va97] Cognition Vander und Technology Group at. *The Jasper Project - Lessons in Curriculum, Instruction, Assessment, and Professional Development*. Routledge, New York, pap/cdr. Auflage, 1997.
- [WBH10] Martin Wagenschein, Hans Ch. Berg und Hartmut von Hentig. *Verstehen lehren - Genetisch - Sokratisch - Exemplarisch*. Beltz, Langensalza, 2. Aufl., Auflage, 2010.
- [Wit03] Helmut Witten. Allgemeinbildender Informatikunterricht? Ein neuer Blick auf H. W. Heymanns Aufgaben allgemeinbildender Schulen. In Peter Hubwieser, Hrsg., *INFOS*, Jgg. 32, Seiten 59–75. GI, 2003.

Debugging-Aufgaben – Programmfehler als Lernchance

Hanno Schauer

Mons-Tabor-Gymnasium
Von-Bodelschwingh-Straße 35
56410 Montabaur
hanno.schauer@uni-duisburg-essen.de

Abstract: Der Beitrag präsentiert einen (neuen) Aufgabentyp für den Unterricht in Algorithmik/Programmierung, die sogenannten Debugging-Aufgaben. In Debugging-Aufgaben wird mithilfe des Debuggers einer Programmierumgebung der Quelltext eines mit Fehlern versehenen Programms analysiert und von Fehlern bereinigt. Hierbei sind die Schülerinnen und Schüler gefordert, sich intensiv und entdeckend mit den Ursachen der Fehler bzw. den zugehörigen (Programmier-) Konzepten auseinanderzusetzen. Der Beitrag integriert Unterrichtserfahrungen an einem rheinland-pfälzischen Gymnasium.

1 Realität und Anspruch in Algorithmik/Programmierung

Die Algorithmik/Programmierung ist ein basaler Baustein jedes Informatikkurses – dies gilt an Schulen ebenso wie an Universitäten. Die Algorithmik/Programmierung ist hierbei konzeptuelles und methodisches Fundament ebenso wie auch Stolperstein: Denn einerseits bauen nachfolgende Reihen auf den Inhalten der Algorithmik/Programmierung auf; dies gilt gleichermaßen für allgemeine Daten- und Kontrollstrukturen wie auch die jeweilige Programmiersprache. Andererseits weist gerade die grundlegende Reihe der Algorithmik/Programmierung eine tendenziell hohe Misserfolgsquote auf [Mo11]: Vornehmliche Ursache des Scheiterns der Schülerinnen oder Schüler ist hierbei die (unzureichende) Beherrschung der konkreten Programmiersprache (oder Notationsform), nicht das Verständnis der algorithmischen Grundstrukturen. Lehrende in der Algorithmik/Programmierung stehen mithin stets vor der Herausforderung zu verhindern, dass die Eigenheiten der jeweiligen Programmiersprache den Unterricht unangemessen dominieren.

Die geschilderte Problematik ist nicht neu und es existierten erprobte didaktische Mittel hierfür. Generell ist der einschlägige Unterricht so zu strukturieren, dass allgemeine Konzepte der Algorithmik einerseits und deren Umsetzung in einer Programmiersprache andererseits für Schülerinnen und Schüler gut unterscheidbar bleiben. Zur Unterstützung speziell der Programmierung existiert darüber hinaus eine ansehnliche Zahl an Softwarewerkzeugen, die darauf abzielen, die Hürden der Programmierung für Anfänger zu reduzieren. Hierbei ist zu denken an (1) Mini-Languages mit eingeschränktem Befehlsatz (z. B. *Robot Karol*¹, *Kara*²), an grafische Programmiersprachen, welche syntaktisch

¹ Siehe: <http://www.schule.bayern.de/karol/>

² Siehe: <http://www.swisseduc.ch/informatik/karatojava/index.html>

fehlerhaften Code von Haus aus nicht ermöglichen (z. B. *BYOB*³), oder an Entwicklungsumgebungen, die eine integrierte Pflege des Quellcodes eines Programms und eines zugehörigen Datenmodells (i. d. R. eines Klassendiagramms) erzwingen (z. B. *BlueJ*⁴) bzw. ermöglichen (z. B. *Java Editor*⁵). Auch Verbindungen der genannten Ansätze sind existieren – bspw. kombiniert *Greenfoot*⁶ Ansätze einer Mini-Language mit der modellbasierten Entwicklung, *Scratch*⁷ wiederum offeriert eine grafisch notierte Mini-Language.

Nachfolgend soll mit den sogenannten Debugging-Aufgaben ein (neuer) Aufgabentyp der Algorithmik/Programmierung vorgestellt werden. Während der Bearbeitung einer Debugging-Aufgabe untersucht der Lerner den Quelltext eines lauffähigen, aber fehlerhaften Programms mithilfe des Debuggers (s)einer Entwicklungsumgebung. Hierbei gilt es, die Fehlerquellen innerhalb des Quelltextes des Programms zu lokalisieren, die Fehler zu analysieren und schließlich zu korrigieren. Im Unterschied zu den eingangs genannten Ansätzen verringern Debugging-Aufgaben nicht die Hürden der Programmierung, sondern leiten Schülerinnen und Schüler dabei an, ausgesuchte Eigenschaften einer Programmiersprache, einer Kontrollstruktur oder einer Datenstruktur entdeckend und so intensiv zu untersuchen, wie es die Behebung des jeweiligen Fehlers erfordert. Debugging-Aufgaben sind kongruent zu den vorgenannten Ansätzen und lassen sich bei Bedarf gut mit diesen kombinieren.

Die Vorstellung der Debugging-Aufgaben selbst erfolgt in Kapitel 3. Dieses vorbereitend gibt Kapitel 2 einen Überblick über Debugging-Tätigkeiten und -werkzeuge. Kapitel 4 evaluiert auf Basis einer Schülerbefragung und Unterrichtserfahrungen des Autors das Konzept der Debugging-Aufgaben. Der Beitrag schließt mit einem Ausblick (Kapitel 5).

2 Debugging und Debugger

Unter Debugging (wörtlich: „entwanzen“) subsumiert die Softwaretechnik Tätigkeiten einer gezielten Fehlersuche und -eliminierung in kompilierten, also grundsätzlich lauffähigen Programmen. Einschlägige Vorgehensweisen sind diesbezüglich (händische) ‚Schreibtischtests‘, das Programmieren von Fehlerabfragen, die Auswertung von Log-Files und der Einsatz dedizierter Debugging-Werkzeuge (z. B. [GB+09], [Ba05]).

Debugging-Tätigkeiten richten sich auf die Eliminierung von Laufzeit- und Semantikfehlern⁸, nicht aber auf Syntaxfehler, da letztere bereits beim Compilieren erkannt werden: Laufzeitfehler verhindern die reguläre Ausführung eines Programms, da sie dieses entweder zum Absturz – bspw. nach Division durch Null – oder zum Halten bringen – z. B. in einer Endlosschleife. Mit Semantikfehlern behaftete Programme hingegen wer-

³ Siehe: <http://snap.berkeley.edu/>

⁴ Siehe: <http://www.bluej.org/>

⁵ Siehe: <http://www.javaeditor.org>

⁶ Siehe: <http://www.greenfoot.org>

⁷ Siehe: <http://scratch.mit.edu/>

⁸ Neben *Semantikfehler* ist auch der Begriff *Logischer Programmfehler* gebräuchlich (z. B. [Ba05, S. 167]).

den performant ausgeführt, das Programmverhalten entspricht aber nicht dem vom Programmierer Gewollten – z. B. indem falsche Werte berechnet werden.

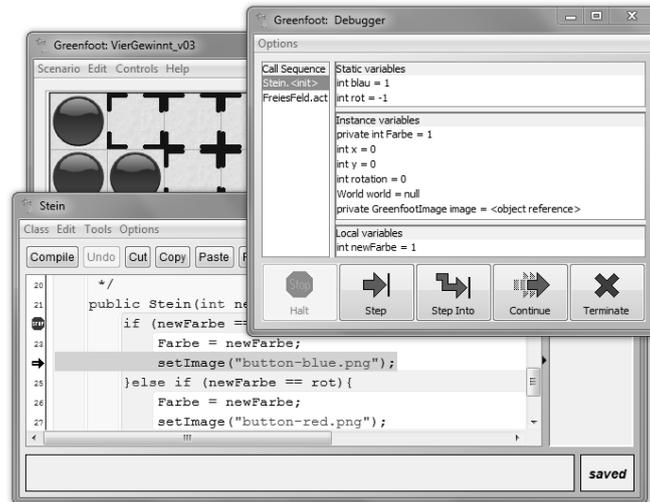


Abbildung 1: Debugging in Greenfoot: Applikation (hinten), Quelltext mit Haltepunkt (Mitte), Debugger mit aktuellen Variablenzuständen und Steuertasten des Tracemodus (vorne)

Werkzeugunterstützung erfährt das Debugging durch dedizierte Debugging-Werkzeuge – häufig in einer Programmoberfläche integriert, dem *Debugger*. Entsprechende Funktionalitäten sind ein üblicher Bestandteil (fast) jeder Programmierumgebung. Moderne Debugger beinhalten auch sehr elaborierte Funktionalitäten – bspw. zur Analyse von Stapelverarbeitung und Funktionsaufrufen, von Speicherabbildern oder Prozessorzuständen. Im und für den Schulunterricht genügt es, sich auf die folgenden drei Kernfunktionalitäten eines Debuggers zu fokussieren: (1) *Haltepunkte* zu setzen, bei deren Erreichung ein laufendes Programm angehalten wird, (2) gestoppte Programme im *Tracemodus* Schritt für Schritt zu steuern, sowie (3) den *internen Zustand* von Variablen und Ausdrücken bzw. dessen Veränderung während der Programmausführung im Tracemodus (quasi live) zu beobachten (vgl. auch Abbildung 1). Die nachfolgenden Ausführungen rekurren (nur) auf die vorgenannten Kernfunktionalitäten.

In der Literatur der Didaktik der Schul informatik ebenso wie in einschlägigen Schulbüchern werden der Debugger bzw. das Debugging eher am Rande behandelt. Einerseits wird die (in der Pädagogik selbstverständliche) Bedeutung einer unterrichtlichen Fehlerkultur und des damit ermöglichten Lernens aus Fehlern auch für die Informatik betont (z. B. [Hu06], [Mo04]). Eine Verbindung der Fehlerkultur mit der Nutzung von Debuggern wird in den diesbezüglichen Quellen nicht hergestellt. Andererseits werden Debugger und Debugging in verschiedenen Quellen als Werkzeug bzw. Technik der Programmierung erwähnt und einführend erläutert (z. B. [En06], [Ha10]). Eine vertiefende Nutzung im Unterricht, so wie sie nachstehend vorgeschlagen wird, findet indes nicht statt.

3 Debugging-Aufgaben

Im Folgenden sollen Debugging-Aufgaben vorgestellt und aufgezeigt werden, wie diese gewinnbringend im Unterricht eingesetzt werden können. Kapitel 3.1 beschreibt diesbezüglich das didaktische Konzept Debugging-Aufgabe. Kapitel 3.2 nennt einschlägige, erfolgreich getestete Anwendungsfelder. Kapitel 3.3 umreißt schließlich auf Unterrichtserfahrungen basierende Regeln guter Praxis für Debugging-Aufgaben.

3.1 Gegenstand

In einer Debugging-Aufgabe gilt es, den Quelltext eines fehlerbehafteten Programmes mithilfe des Debuggers einer (der gewohnten) Programmierumgebung zu entdecken und zu korrigieren. Die Lerner müssen den oder die Fehler im Quelltext lokalisieren, analysieren und bereinigen. Das Vorgehensmodell einer Debugging-Aufgabe ist diesbezüglich wie folgt:

- 1) **Fehler aufdecken / rekonstruieren:** Durch das testweise Anwenden des Programms der Debugging-Aufgabe rekonstruieren die Schülerinnen und Schüler das in der Aufgabenstellung beschriebene bzw. angedeutete Fehlerverhalten des Programms und grenzen es ggf. auf bestimmte auslösende Bedingungen (z. B. bestimmte Benutzereingaben) ein.
- 2) **Fehlerquelle lokalisieren:** Ab einem zuvor gesetzten Haltepunkt werden das Programm im Tracemodus solange Schritt für Schritt durchlaufen und die Werte relevanter Variablen überwacht, bis eine Fehlfunktion des Programms (Laufzeitfehler) oder unpassende Variablenwerte (Semantikfehler) beobachtet und die auslösende, fehlerhafte Anweisung damit lokalisiert werden kann.
- 3) **Fehlerhaften Quelltext analysieren:** Nun gilt es, die Stelle(n) des Quelltextes, an der/denen ein fehlerhaftes Programmverhalten beobachtet wurde, so intensiv zu auf Fehler und deren Ursachen zu analysieren, dass eine Korrektur des oder der Fehler möglich wird.
- 4) **Fehler korrigieren:** Schließlich ist der Quellcode so abzuändern, dass der oder die vorliegende(n) Fehler nicht mehr auftreten.

Obiges Vorgehensmodell ist gegebenenfalls mehrmals zu durchlaufen, insbesondere bei mehreren Programmfehlern. Rücksprünge zu vorhergehenden Phasen innerhalb des Vorgehensmodells sind nicht nur möglich, sondern üblich.

Debugging-Aufgaben erfordern die Einarbeitung in einen fremderstellten, teils nur eingeschränkt lauffähigen Quelltext (Laufzeitfehler). Es ist deshalb vielfach sinnvoll, den Schülerinnen und Schülern zusätzlich das Programm erläuternde Materialien an die Hand zu geben – beispielsweise konzeptuelle Modelle (z. B. Use Case- und Klassendiagramme), kompilierte Musterlösungen des Programms oder, bei realitätsnah gehaltenen Aufgaben, auch eine strukturierte Anforderungsspezifikation (siehe auch Kapitel 3.2.3).

Der Quelltextumfang einer Debugging-Aufgabe ist typischer Weise deutlich größer (Faktor 2 bis 3) als die Menge Code, die ein durchschnittlicher Lerner innerhalb einer Schulstunde selbst zu verfassen vermag. Hierbei gilt, dass je übersichtlicher, je strukturierter und je besser dokumentiert der Quelltext einer Debugging-Aufgabe ist, desto umfangreicher kann er tendenziell gestaltet werden (siehe auch Kapitel 3.3).

```
29 // Nullstellen berechnen
30 double diskriminante = Math.sqrt((p/2)*(p/2)-q);
31 if (diskriminante >= 0) {
32     x1 = p/2 + diskriminante;
33     x2 = p/2 - diskriminante;
34 }
```

Abbildung 2: Ausschnitt aus dem (fehlerhaften) Quelltext einer Debugging-Aufgabe

Das Beispiel in Abbildung 2 zeigt einen Ausschnitt einer in das Thema einführenden Debugging-Aufgabe. Es sollen Nullstellen einer quadratischen Funktion mithilfe der p-q-Formel berechnet werden. Allerdings wurden im Quelltext zwei Fehler untergebracht, ein Laufzeit- und ein Semantikfehler. Beim ersten Testen des Programms würde das Programm regelmäßig abstürzen. Nach Setzen eines Haltepunkts in Zeile 30 und Ausführung dieser Zeile im Tracemodus würde die Quelle des Laufzeitfehlers (hier: Wurzel aus einer negativen Zahl) ebenfalls in Zeile 30 entdeckt werden. Diese könnte dann – z. B. durch das Ziehen der Wurzel erst in den Zeilen 32 ff. – korrigiert werden. Ein erneuter Programmtest würde nun den Semantikfehler in Form falsch berechneter Nullstellen offenbaren. Auch dieser wäre schnell gefunden; schon die Betrachtung der p-q-Formel nämlich offenbart ein fehlendes Minus jeweils vor „p/2“ (Zeilen 32 und 33).

3.2 Anwendungsgebiete

Debugging-Aufgaben empfehlen sich in mehrfacher Hinsicht für den Unterricht. Sie unterstützen bei der Vermittlung der Programmier- und Debugging-Technik (Kapitel 3.2.1). In der Algorithmik/Programmierung leiten sie Schülerinnen und Schüler bei der Analyse von Eigenheiten und Fehlerquellen einer Programmiersprache bzw. ausgewählter algorithmischer Konzepte an (Kapitel 3.2.2). In der Softwaretechnik sind sie ein empfehlenswerter Aufgabentyp für die Phasen Test und Wartung des Softwarelebenszyklus (Kapitel 3.2.3). Zudem haben sie sich als knifflige Zusatzaufgaben bewährt (Kapitel 3.2.4).

3.2.1 Debugging-Kenntnisse vermitteln

Bevor der Debugger im Unterricht genutzt werden kann, gilt es, den Schülerinnen und Schülern dessen Kernfunktionalitäten näherzubringen – also Haltepunkte, Tracemodus, Variablenbelegung (siehe auch Kapitel 2). Für die Vermittlung von Debugging-Tätigkeiten als solches empfehlen sich (einfache) Anwendungsaufgaben für den Debugger, sprich (einfache) Debugging-Aufgaben. Ein geeigneter Zeitpunkt für eine entsprechende Unterrichtseinheit ergibt sich nach Erfahrung des Autors dann, wenn alle grundlegenden Kontrollstrukturen der Programmierung behandelt worden sind. Der Umfang einer diesbezüglichen Einheit ist mit zwei bis vier Unterrichtsstunden überschaubar.

Es sei angemerkt, dass eine Einführung in das Debugging als wichtige Programmier-technik auch dann sinnvoll ist, wenn im weiteren Unterrichtsgang Debugging-Aufgaben nicht genutzt werden sollten.

3.2.2 Fehlerquellen erkennen

Debugging-Aufgaben führen Schülerinnen und Schüler in einem entdeckenden Lernprozess an die im Quelltext der Aufgabe untergebrachten Fehlerquellen; diese sind dann solange zu untersuchen, bis es gelingt den (verstandenen) Fehler zu korrigieren. Diesbezüglich sind Debugging-Aufgaben gut geeignet, um Eigenheiten und typische Fehlerquellen ausgesuchter Programmierkonzepte durch Schülerinnen und Schüler entdecken und ergründen zu lassen. Im Unterricht des Autors wurden Debugging-Aufgaben unter anderem erfolgreich eingesetzt, um den Unterschied von Deklaration und Initialisierung einer Variable herauszuarbeiten, um den (fehlschlagenden) Zugriff auf nicht vorhandene Ressourcen (Elemente eines Arrays, Dateien) zu demonstrieren und, um rekursive Funktionsaufrufe durch Schülerinnen und Schüler (entdeckend) nachvollziehen zu lassen.

Testweise gelang es im Unterricht des Autors sogar, Schülerinnen und Schüler eines Informatik-Grundkurses die softwaretechnisch komplexen Fehlerquellen von Rollenbeziehungen in Klassenhierarchien (vgl. [Fr00]) sowie der Ko- und Kontravarianz (vgl. [Pr01]) exemplarisch aufdecken und ergründen zu lassen – ohne Fehlerkorrektur allerdings.

3.2.3 Test und Wartung in der Softwareentwicklung

In einer Reihe zur Softwaretechnik haben Schülerinnen und Schüler aufgrund ihrer Vorerfahrungen in der Programmierung i. d. R. ein gutes Verständnis für die Herausforderungen und Tätigkeiten der Planung und Entwicklung von Software. Die Bedeutung des Testens und der (langfristigen) Wartung von Software ist ihnen intellektuell meist ebenfalls präsent, allerdings fehlt nicht selten, ein implizites, erfahrungsbasiertes Verständnis dieser Phasen. Hier bewähren sich Debugging-Aufgaben in zweifacher Weise. Im Vorfeld der Reihe erzeugen sie praktische Erfahrungen im Testen und Pflegen von Software. Während der Reihe selbst bieten sich Debugging-Aufgaben als praktische Übungen in den entsprechenden „Phasen“ an. Aus Sicht des Autors sind hierbei anspruchsvolle Debugging-Aufgaben empfehlenswert, welche an den Original-Kontext des Software-Testens anknüpfen – z. B. mit Blick auf die Einhaltung einer gegebenen Softwarespezifikation oder die Bearbeitung einer fiktiven Fehlermeldung („Trouble Ticket“).

3.2.4 Zusatzaufgaben

Debugging-Aufgaben erfordern die Beschäftigung mit fremdem Quellcode, weshalb sie sich fast beliebig zeitintensiv gestalten lassen. Die Besprechung der Aufgaben lässt sich dagegen meist in kurzer Zeit erledigen – schließlich gilt es (im Wesentlichen) nur, den oder die Fehler aufzuzeigen. Hierbei ist das Erklären eines Fehlers i. d. R. auch für Schülerinnen und Schüler nachvollziehbar, wenn sie die Aufgabe nicht oder nicht vollständig bearbeitet haben. Diese Eigenschaft empfiehlt Debugging-Aufgaben – auch aus der

Erfahrung des Autors heraus – in besonderer Weise als Zusatzaufgaben – z. B. um schnelle Programmierer im Unterricht sinnvoll zu beschäftigen.

3.3 Empfehlungen guter Praxis

Wie bei (fast) allen didaktischen Instrumenten, hängt auch bei Debugging-Aufgaben der (Bildungs-) Erfolg nicht zuletzt von der gelungenen Gestaltung und dem angemessenen Einsatz des Instruments ab. Einschlägige Unterrichtserfahrungen des Autors sind diesbezüglich in den nachfolgenden Regeln guter Praxis zusammengefasst und ergänzen die Evaluation des Konzepts (siehe Kapitel 4).

Der richtige Schwierigkeitsgrad

Das Austarieren des Schwierigkeitsgrads einer Debugging-Aufgabe ist nicht trivial: Aufgaben, bei denen es Schülerinnen und Schülern nicht gelingt, einen Fehler (in absehbarer Zeit) zu analysieren, führen kaum zu Lernerfolg, eher zu Frustration. Aufgaben mit geringer Komplexität wiederum werden vielfach ohne Debugger und ohne intensivere Fehleranalyse gelöst – z. B. durch simples Ausprobieren. Als Faustregeln hat es sich diesbezüglich bewährt, Quelltexte in der zwei- bis dreifachen Länge dessen zu nutzen, was eine mittelschnelle Schülergruppe⁹ in einer Unterrichtsstunde typischer Weise zu verfassen vermag. Andererseits hat es sich als sinnvoll herausgestellt, nur solche Quelltexte in einer Debugging-Aufgabe zu nutzen, bei denen auch ein erfahrener Programmierer einen Debugger einsetzen würde.

Bezüglich der Ausrichtung des Schwierigkeitsgrades ist ebenfalls zu beachten, dass Schülerinnen und Schüler durch ihr mathematisches Vorwissen mathematische Laufzeitfehler (Division durch Null, Wurzel aus einer negativen Zahl) gut verstehen und schnell erkennen. Entsprechende Aufgaben bieten sich somit vornehmlich für den Einführungsunterricht in das Debugging an.

Gestaffelte Fehler in gestaffelter Schwierigkeit

Im Unterricht des Autors differierte die Bearbeitungszeit für Debugging-Aufgaben zwischen einzelnen Schülergruppen häufig sehr stark – dem Eindruck nach deutlicher als bei Programmieraufgaben. Diesbezüglich zeigten sich solche Debugging-Aufgaben als besonders geeignet, die mehrere Fehler in gestaffelter Schwierigkeit enthielten – z. B. ein Laufzeitfehler und ein Semantikfehler. Bei entsprechenden Aufgaben zeigte es sich als wahrscheinlich, dass auch langsamere Aufgabenlöser Teilergebnisse bereits erreichen, noch bevor die schnelleren Löser die Aufgabe als Ganzes bewerkstelligen. Gleichzeitig wurde aus Schülerrückmeldungen deutlich, dass Schülerinnen und Schüler Debugging-Aufgaben besonders dann als sinnvollen Lerngegenstand erachten, wenn während der Aufgabe ein steter Lösungsfortschritt klar zu erkennen ist. Umgekehrt verunsicherten Aufgaben mit logisch ineinander verwobenen Programmfehlern insbesondere weniger programmiersichere Schülerinnen und Schüler merklich.

⁹ Im Unterricht des Autors arbeiten Schülerinnen und Schüler ausschließlich in Kleingruppen am PC.

Struktur – Struktur – Struktur

Das Einarbeiten in fremden Quellcode gelingt grundsätzlich leichter, wenn dieser wohl strukturiert und modularisiert ist, ausführlich kommentiert wurde und gut leserlich implementiert ist. Ergänzende Formen der Dokumentation, insbesondere konzeptuelle Modelle wie Schnittstellenbeschreibungen, Use Case- oder Klassendiagramme unterstützen die Einarbeitung zusätzlich. Es empfiehlt sich diesbezüglich in mehrfacher Hinsicht, Schülerinnen und Schülern in Debugging-Aufgaben wohl strukturierte Quelltexte und zusätzliche Dokumentation zu Verfügung zu stellen: Erstens, weil dadurch die Bearbeitung der Aufgabe erleichtert wird. Zweitens zeigte sich, dass je leichter verständlich ein Quelltext gestaltet war bzw. je zugänglicher er durch begleitende Dokumente wurde, desto umfangreicher konnte er sein, so dass auch größere Programmtexte regelmäßiger Teil des Informatikunterrichts werden können. Drittens erfahren Schülerinnen und Schüler den Nutzen von Struktur, Modellierung und Dokumentation insbesondere dann sehr deutlich, wenn sie, wie in Debugging-Aufgaben, in einer direkten Weise hiervon abhängig sind bzw. hiervon profitieren.

Klare Arbeitsaufträge

Auch im Debuggen geübte Schüler neig(t)en zuweilen dazu, die im Quelltext versteckten Fehler ohne den Debugger finden und korrigieren zu wollen, was bei umfangreicheren Debugging-Aufgaben i. d. R. nicht gelang und die eingesetzte Lernzeit somit weitgehend unproduktiv verging. Es hat sich diesbezüglich als sinnvoll und hinreichend herausgestellt, Debugging-Aufgaben klar als solche zu betiteln und in diesen – auch gegenüber damit vertrauten Schülern – dezidiert die Verwendung des Debuggers einzufordern.

4 Evaluation und Einordnung

Debugging-Aufgaben sind keine gesonderte didaktische Methode oder ein eigenständiges Unterrichtskonzept, sondern (nur) ein spezieller Aufgabentyp, der an geeigneter Stelle in den Unterricht integriert wird (siehe hierfür Kapitel 3.2). Mit Ausnahme der Vermittlung des Debuggings als solches bestimmen das Debugging bzw. Debugging-Aufgaben keine Unterrichtseinheit, sondern ergänzen diese. Gleichzeitig sind Debugging-Aufgaben aktuell noch in einer – wenn auch gereiften – Entwicklungsphase. Vor diesem Hintergrund erfolgte die Evaluation der Debugging-Aufgaben nicht durch eine statistische Messung des durch Debugging-Aufgaben erzeugten Lernerfolges – z. B. durch die vergleichende Evaluation zweier Lerngruppen –, sondern anhand einer Schülerbefragung und der (qualitativen) teilnehmenden Beobachtung durch den Autor.

Die hier grundgelegten Unterrichtserfahrungen entstammen drei Informatikgrundkursen mit 11 (Kurs A), 16 (Kurs B) bzw. 23 Schülerinnen und Schülern (Kurs C), in denen der Autor Debugging-Aufgaben einsetzte und testete: Alle Kurse erhielten eine Einführung in das Debugging (siehe Kapitel 3.2.1) und hatten Debugging-Aufgaben im weiteren Fortgang der Algorithmik/Programmierung zu bearbeiten (siehe Kapitel 3.2.2 und 3.2.4). Nur ein Kurs, Kurs A, bearbeitete Debugging-Aufgaben bislang in der Softwaretechnik-Reihe (siehe Kapitel 3.2.3).

In allen drei Kursen fand eine Schülerbefragung statt, in welcher die Schüler aufgefordert waren, ihre Erfahrungen mit Debugging-Aufgaben in wenigen Worten zu formulieren. Das Feedback des ersten Kurses (Kurs A) floss direkt in die Verbesserung derjenigen Aufgaben ein, die den Folgekursen präsentiert wurden. Aus den Rückmeldungen der beiden weiteren Kurse wurden – mit Blick auf den vorliegenden Beitrag – die folgenden fünf Aussagen im Plenum ausgewählt, welche nach Ansicht der Anwesenden geeignet sind, die geäußerten Argumente prägnant zu verdichten:

„Man weiß, wie man sich selbst helfen kann.“

„Einige kritische Fehlerquellen wurden mir besser verständlich.“

„Fehlererkennen ist noch nicht das Fehlerbeseitigen. Dies ist eine zusätzliche Schwierigkeit.“

„Das Lesen und Verstehen fremder Programme war eine ganz eigene, neue Herausforderung“

„Debugging-Aufgaben bringen Abwechslung in den Unterricht. Selbst programmieren ist aber interessanter.“

Einschlägige Unterrichtserfahrungen des Autors flossen bereits in die Kapitel 3.2 und 3.3 in Form von sinnvollen *Anwendungsgebieten* bzw. *Empfehlungen guter Praxis* für Debugging-Aufgaben ein. Die dort aufgeführten Eigenschaften von Debugging-Aufgaben bzw. des zugehörigen Unterrichts zählen zu den Inhalten einer Evaluation des Konzeptes. Sie sollen aber an dieser Stelle nicht wiederholt werden. Der Leser wird stattdessen gebeten, den Inhalt der beiden genannten Kapitel in seine Gesamtwürdigung mit einzubeziehen. Im Folgenden werden ergänzend noch zwei (weitere) prägende Merkmale von Debugging-Aufgaben erläutert, die sich nicht in die Inhalte der vorstehenden Kapitel integrierten:

Klarer Fokus auf Details: Debugging-Aufgaben befördern auf Seiten der Schülerinnen und Schüler entdeckendes Lernen. Der Lösungsweg der Aufgaben ist i. d. R. durch die gesetzten Fehler und die Funktionsweise des Debuggers aber eng vorgegeben. En gros sind Debugging-Aufgaben also keine offenen Aufgaben. Sie fokussieren vielmehr auf konkrete, identifizierbare Fehler und den diesbezüglichen Programmausschnitt. Intensiv analysiert werden hierbei in aller Regel (nur) die fehlerhaften Quelltextausschnitte.

Nicht unbeträchtlicher Erstellungsaufwand: Die Erstellung der Debugging-Aufgaben des Autors war – verglichen mit gängigen Programmieraufgaben – mit deutlich erhöhtem Aufwand verbunden. Denn es galt, zuerst ein fehlerfreies Programm sowie begleitende Dokumentation zu erstellen, wobei der Quelltextumfang denjenigen typischer schulischer Programme merklich übersteigt (vgl. Kapitel 3.1). Erst dann wurden die für den Unterricht relevanten Fehler im Quelltext verankert. Die Erstellung einiger, vom Autor genutzter Debugging-Aufgaben nahm jeweils mehr als einen Arbeitstag in Anspruch.

5 Fazit und Ausblick

Debugging-Aufgaben haben sich im Unterricht des Autors bewährt; sie sind dort mittlerweile fester Bestandteil. Sie haben sich hier als (sinnvolle) Ergänzung all der Unterrichtsreihen gezeigt, in denen intensiv programmiert wird – im Unterricht des Autors sind dies die Algorithmik/Programmierung, die Softwaretechnik und die (ebenfalls programmierintensiv gehaltene) Kryptologie.

Die geschilderten Vorteile von Debugging-Aufgaben verbinden sich mit der deutlichen Herausforderung eines teils sehr beträchtlichen Erstellungsaufwands. Diesbezüglich ist es weder sinnvoll, noch erwartbar, Debugging-Aufgaben von jeder Informatiklehrkraft allein kreieren zu lassen. Hier ist vielmehr die Informatik-Didaktik-Community gefordert, den Praktikern vor Ort qualitätsgesicherte und gut dokumentierte Aufgaben zu Verfügung zu stellen. Einen ersten Schritt in diese Richtung plant der Autor selbst. Ein an Informatiklehrer und -lehrerinnen gerichteter Artikel, der die Anwendung von Debugging-Aufgaben im Unterricht anhand von Beispielen praxisnah aufzeigt, ist in Vorbereitung. Parallel hierzu will der Autor ausgewählte Debugging-Aufgaben online zu Verfügung stellen.

Literaturverzeichnis

- [Ba05] Balzert, H.: Lehrbuch Grundlagen der Informatik. Elsevier, München, 2005.
- [En06] Engelmann, L (Hrsg.); Bartke, P., Burkhard, H.-D. et al.: Informatik – Gymnasiale Oberstufe. Duden, Berlin, Frankfurt/Main, 2006.
- [Fr00] Frank, U.: Delegation: An Important Concept for the Appropriate Design of Object Models. JOOP – Journal of Object-Oriented Programming. 13 (2000) 3, S. 13-18.
- [GB+09] Grechenig, T.; Bernhart, M.; Breiteneder, R.; Kappel, K.: Softwaretechnik. Pearson, München et al., 2009.
- [Ha10] Hattenauer, R.: Informatik für Schule und Ausbildung. Pearson, München et al., 2010.
- [Hu06] Humbert, L.: Didaktik der Informatik – mit praxiserprobtem Unterrichtsmaterial. Teubner, Wiesbaden, 2006.
- [Mo04] Modrow, E.: Einführung in die Informatik – Teil 2 – Java-Anweisungen und -Ausdrücke. Virtuelle Lehrerweiterbildung Informatik in Niedersachsen, Scheden, 2004. [http://www.vlin.de/material_2/Struktogramme-Java.pdf, 03.03.2013]
- [Mo11] Modrow, E.: Visuelle Programmierung – oder: Was lernt man aus Syntaxfehlern? Thomas, M. (Hrsg.): Informatik in Bildung und Beruf (INFOS 2011): Gesellschaft für Informatik, Bonn, 2011, S. 27-36.
- [Pr01] Prasse, M: Entwicklung und Formalisierung eines objektorientierten Sprachmodells als Grundlage für MEMO-OML. Fölbach, Koblenz, 2001.

Informatik und Mathematik – kombiniert im Schülerlabor- Modul „Einstieg in die Computergrafik“

Nadine Bergner, Tim Schellartz, Ulrik Schroeder

Lehr- und Forschungsgebiet Informatik 9
RWTH Aachen
Ahornstr. 55
52074 Aachen
bergner@informatik.rwth-aachen.de
schroeder@informatik.rwth-aachen.de
tim.schellartz@rwth-aachen.de

Abstract: Das Modul „Wie kommt das Bild auf den Bildschirm? - Einstieg in die Computergrafik“ bietet informatikbegeisterten Schülerinnen und Schülern (SuS) der Sekundarstufe II die Möglichkeit bereits vor Studienbeginn experimentell zu erfahren, warum und inwiefern die Mathematik in der Informatik eine wesentliche Rolle spielt. Im Rahmen eines sechsständigen Moduls werden unter anderem die Inhalte Objektdarstellung mittels Polygonnetzen, Transformation von Objekten, Winkelbestimmung zwischen zwei Ebenen, RGBA-Farbsystem, Phong-Beleuchtung und 3D-Stereosehen von den SuS selbstständig erforscht. Im Folgenden wird aufbauend auf der Motivation für die Thematik Computergrafik für SuS und der Analyse der theoretischen Grundlagen das Modul insgesamt dargestellt und anschließend die didaktische und methodische Ausgestaltung detailliert beleuchtet.

1 Einleitung und Motivation

Die Antwort auf die Frage, was genau ein Informatikstudium umfasst, fällt vielen SuS auch kurz vor dem (Fach-)Abitur noch schwer (vgl. [ELK10], [He11]). Darüber hinaus haben die wenigsten Abiturienten und auch nicht alle Studienanfänger eine Vorstellung davon, warum der Mathematik gerade in den Grundvorlesungen eine so große Bedeutung zukommt (vgl. [MW06]). Um diesen und anderen Fehlvorstellungen frühzeitig entgegen zu wirken, bietet das InfoSphere – Schülerlabor Informatik¹ der RWTH Aachen für SuS der Sekundarstufe II das Modul „Wie kommt das Bild auf den Bildschirm? - Einstieg in die Computergrafik“ an. Das Thema Computergrafik eignet sich hervorragend, um die SuS in ihrer Lebenswirklichkeit abzuholen und ihnen gleichzeitig einen realistischen Einblick in ein Informatikstudium zu ermöglichen.

Computergrafik findet man heute überall, so zum Beispiel bei vielen Spezialeffekten im Kino, der eingeblendeten Wetterkarte in den Nachrichten oder auch bei animierten

¹ <http://schuelerlabor.informatik.rwth-aachen.de>

Hintergründen im Smartphone. Neben dem motivierenden Alltagsbezug ermöglicht es die Computergrafik auf anschauliche Weise die starke Verknüpfung der Mathematik mit der Informatik bewusst zu machen.

2 Hintergründe und Related Work

Auch 2013 ist das Schulfach Informatik an weiterführenden Schulen in NRW immer noch stark unterrepräsentiert und wird höchstens in Form von ITG-Kursen (Informationstechnische Grundbildung) in der Unterstufe, Wahlpflichtkursen in der Mittelstufe und als Grundkurs (selten auch als Leistungskurs) in der Oberstufe angeboten (siehe [Re10]). Diese Situation erfordert es, SuS auch außerhalb des Schulunterrichts auf diese Fachwissenschaft aufmerksam zu machen, da der Informatik heute und in Zukunft eine große Bedeutung zukommt. Dazu wurde 2010 an der RWTH Aachen das InfoSphere – Schülerlabor Informatik gegründet. Im Rahmen dieses außerschulischen Lernorts werden aktuell 23 verschiedene Module für SuS aller Schulstufen angeboten, die den Kindern und Jugendlichen einen ersten oder auch vertiefenden Einblick in die Welt der Informatik ermöglichen. Das Lernen an außerschulischen Lernorten ist bereits seit 1994 Thema der fachdidaktischen Forschung (siehe [Gr94], [Eu05] und [Gu07]). In diesen Arbeiten und weiteren informatisch-didaktischen Forschungen (siehe [Be06], [En04] und [GP08]) kristallisierten sich das aktive, selbstgesteuerte und selbstentdeckende Lernen als eine erfolgreiche Methode für das Informatiklernen heraus, nicht nur in außerschulischen Lernorten. Für eine tiefergehende Analyse der didaktischen Ausgestaltung des InfoSphere sei auf [BHS12] verwiesen. Die Leitidee des selbstentdeckenden Lernens verfolgt auch das hier beschriebene Modul zur Einführung in die Computergrafik.

Neben der fachdidaktischen Forschung gibt es bereits verschiedene Ansätze das Thema Computergrafik für SuS didaktisch aufzuarbeiten, um erste Schritte auf diesem Feld zu ermöglichen und bei den Lernern ein grundlegendes Verständnis zu entwickeln. Als einen Ansatz ist als erstes die Website NeHe² zu nennen, welche mehrere sehr umfangreiche Tutorials zum Selbststudium der plattformunabhängigen Programmiersprache OpenGL Shading Language anbietet. Da diese Website nicht nur Einblicke gewährt, sondern mehrere Wochen umfassende Kurse anbietet, eignet sie sich sehr gut, interessierten SuS nach dem Modul eine eigenständige Vertiefungsmöglichkeit zu ermöglichen und wird daher gerne weiter empfohlen. Des Weiteren gibt es eine von Gabriel Zachmann entwickelte Ideensammlung „Computergrafik in Schule“³, deren Inhalte ebenfalls weit über die des Moduls hinausgehen. Da die Ideensammlung jedoch nur Inhalte und keinerlei didaktische Konzepte beinhaltet, ist diese höchstens engagierten Lehrkräften für den Aufbau einer eigenen Unterrichtssequenz zu empfehlen. Darüber hinaus existiert mit dem Shader-Maker (siehe [Un12]) bereits ein Programm zur Erstellung und Testung von Shaderprogrammen, welches als Vorbild des hier verwendeten Shadermanagers (für detaillierte Beschreibungen siehe Abschnitt 4.2 Didaktische und methodische Entscheidungen) diene.

² <http://nehe.gamedev.net/>

³ http://zach.in.tu-clausthal.de/cg_in_schule/

3 Überblick über das Modul

3.1 Rahmenbedingungen und Vorwissen

Da dieses Modul insbesondere Studieninteressierten verdeutlichen soll welche Rolle die Mathematik in der Fachdisziplin Informatik und damit auch im Informatikstudium spielt, richtet es sich vorrangig an SuS des zweiten und dritten Oberstufenjahres (entspricht den Schulstufen Q1 und Q2 in NRW). Der zeitliche Umfang dieses Moduls liegt bei 6 Zeitstunden inkl. Pausen, was jedoch abhängig vom Vorwissen und Können der Lerngruppe variieren kann.

Entsprechend der Zielsetzung setzt das Modul neben grundlegenden Programmierkenntnissen (bspw. in Java) auch Vorwissen aus der Mathematik voraus. Dies betrifft insbesondere die Vektorrechnung (u.a. Bestimmung von Normalenvektoren, Berechnung von Skalarprodukten) und den sicheren Umgang mit Matrizen (u.a. Multiplikation von Matrizen). Da diese informatischen und mathematischen Vorkenntnisse nur auf niedrigem Niveau benötigt werden, sind sie bereits durch Grundkurse abgedeckt.

3.2 Inhalte des Moduls

Das Modul bietet einen ersten Einstieg in das weite Feld der Computergrafik. Ohne allzu weit in die Tiefe einzudringen, ist es dennoch möglich, dass am Ende des Moduls ein echtes, vollständiges Ergebnis in Form einer animierten Szene entsteht, welches im Anschluss weiter verfeinert, vertieft oder ausgebaut werden kann. Folgende Themen werden (in dieser Reihenfolge) behandelt:

1. Objektdarstellung mittels Polygonnetzen
2. Transformation der erstellten Objekte mittels Matrix-Vektor-Rechnung
3. Winkelbestimmung zwischen zwei Ebenen über das Skalarprodukt der Normalenvektoren
4. RGBA-Farbsystem
5. Phong-Beleuchtung
6. 3D-Stereosehen

Im Folgenden wird das gesamte Modul inklusive Ablauf, Materialien und Software detailliert beschrieben und wichtige didaktische Entscheidungen werden dargelegt.

4 Ausgestaltung des Moduls

4.1 Ablauf

Ziel des Moduls ist eine animierte Szene, die für alle Teilnehmergruppen individuell unterschiedlich und somit einzigartig ist. Zunächst werden Objekte, abstrahiert durch Polygonnetze, mathematisch mittels der Vektoren ihrer Eckpunkte dargestellt und mit

Hilfe eines Steckbrettes und Gummibändern sichtbar gemacht. Daran anknüpfend lernen die SuS verschiedene Möglichkeiten der Transformation (Skalierung, Rotation und Translation) dieser (auch 3-dimensionalen) Objekte mit Hilfe eines speziell dazu entwickelten Applets kennen. Nachdem die Grundlagen in einem Sicherungsblatt gefestigt wurden, erhalten die SuS die Möglichkeit, mit dem Programm „Shadermanager“ erste kleine Vertexshader (Programme zur Veränderung der Form, Farbe und Position der Eckpunkte eines Objekts) zu erstellen. Neben der Farbgebung, die mittels einer kleinen elektrischen Schaltung erlernt wird, animieren die SuS das Objekt mittels den zuvor erlernten Transformationen. Nach Abschluss des Themas Vertexshader folgt die Erarbeitung des Fragmentshaders (Programm zur Veränderung der Oberfläche der Objekte). Um die Objekte möglichst realistisch darzustellen, erwerben die Teilnehmenden ein Grundwissen über lokale Beleuchtung, welches anschließend im Fragmentshader praktisch umgesetzt wird. Darauf aufbauend haben die SuS Gelegenheit, ihre Szene nach eigenen Vorstellungen weiter auszubauen und zu verfeinern. Als Exkurs besteht darüber hinaus die Möglichkeit, auch 3D-Effekte kennenzulernen. Zum Abschluss des Moduls können die Teilnehmenden ihre individuellen Ergebnisse präsentieren und über ihre Umsetzung diskutieren. Damit erhalten die SuS einen umfassenden ersten Einblick in die Welt der Shaderprogrammierung, als einen zentralen Bereich der Computergrafik.

Der zeitliche Ablauf inklusive der jeweiligen Lernmethode und der verwendeten Medien bzw. Materialien kann der folgenden Tabelle entnommen werden.

Zeit	Inhalt	Methode	Medien/Materialien
0.00	Begrüßung und Motivation	Präsentation	Smartboard
0.10	Einstieg in Phase 1: Objektdarstellung	Partnerarbeit	Steckbrett, Arbeitsblatt 1
0.40	Phase 1: Koordinatentransformation	Partnerarbeit	Applet „Koordinaten- transformation“
1.20	Sicherung der 1. Phase	Partnerarbeit	Arbeitsblatt 2
1.40	Einstieg in Phase 2: Präsentation des Programms: Shadermanager	Präsentation	Smartboard
2.00	Phase 2: Animation durch Koordinatentransformationen	Partnerarbeit	Arbeitsblatt 3, Programm „Shadermanager“
2.40	PAUSE	----	----
3.00	Phase 3: RGD-Farbdarstellung	Partnerarbeit	Arbeitsblatt 4, elektr. Schaltung
3.20	Phase 4: Comiclook	Partnerarbeit	Arbeitsblatt 5
4.00	PAUSE	----	----
4.15	Phase 5: Lokale Beleuchtung	Partnerarbeit	Arbeitsblatt 6
4.50	Exkurs: 3D-Stereo	Partnerarbeit	Arbeitsblatt 7, rot-türkis-Brille
5.30	Abschluss: Vorstellung der Ergebnisse	SuS- Präsentation	Smartboard
6:00	ENDE	----	----

4.2 Didaktische und methodische Entscheidungen

Das gesamte Modul wurde der Leitidee des InfoSphere folgend nach dem Prinzip des selbstentdeckenden Lernens gestaltet (vgl. [BHS12]). Dies bedeutet, dass insbesondere die grundlegenden Themen Objektdarstellung, Koordinatentransformation und RGB-Farbdarstellung auf anschauliche und experimentelle Art und Weise von den Lernern selbst entdeckt werden. Für den Einstieg in die Objektdarstellung wird zur Veranschaulichung ein einfaches Steckbrett mit Gummibändern genutzt; zur Übertragung auf 3-dimensionale Objekte und deren Transformationen dient ein Applet, welches speziell auf die Bedürfnisse der Zielgruppe zugeschnitten ist; und die RGB-Farbdarstellung können die SuS lernerzentriert mittels einer elektronischen Schaltung selbst erfahren.

Ein weiteres grundsätzliches Merkmal der Phasen 2 bis 5 ist das Prinzip der abnehmenden Hilfestellung und schrittweisen Öffnung der Arbeitsaufträge (vgl. [MN11]). Die relativ kleinschrittige Anleitung in Phase 2 stellt sicher, dass alle SuS mit ersten Erfolgserlebnissen in der Shaderprogrammierung starten. Anschließend bieten die Aufträge immer mehr Spielraum zur Entfaltung der eigenen Ideen.

Im Folgenden werden die wichtigsten didaktischen und methodischen Entscheidungen der einzelnen Phasen genauer erläutert.

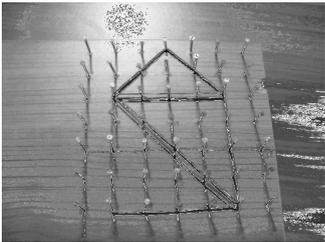
Einstieg in Phase 1: Objektdarstellung mittels Polygonnetzen

Als Einstieg in das Modul entdecken die SuS selbstständig den Zusammenhang zwischen einfachen Strich-Grafiken (Polygonnetzen) und den zugrundeliegenden mathematischen Konstrukten (Vektoren und Geraden), indem sie einfache Dreiecksflächen mithilfe gegebener Vektoren auf einem Nagelbrett abstecken (siehe dazu Abbildung 1). Diese recht einfache praktische Aufgabe ermöglicht es den SuS im Modul anzukommen, was wichtig für weitere Lernphasen ist, da die Moduldurchführungen außerhalb des regulären Schulunterrichts stattfinden (siehe [Gr94]).

Objektdarstellung

In der Computergrafik werden Objekte häufig als Netze, auch **Mesh** genannt, dargestellt. Ein Netz besteht aus einer Menge von **Punkten** im Raum, also Vektoren, die durch **Kanten** miteinander verbunden werden. Dies können Dreiecksflächen sein, aber auch andere Vielecke, sprich Polygone, sind möglich.





Aufgabe 1
Steckt auf dem Brett folgende Dreiecksflächen ab. Welches Objekt entsteht?
a) $\begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, b) $\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}$ c) $\begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}$ _____

Abbildung 1: Aufgabe zur Objektdarstellung und zugehörige Lösung

Zur direkten Sicherung der Vektordarstellung können sich die SuS nun eigene Objekte ausdenken, durch Vektoren darstellen und diese mit dem Nachbarsteam austauschen. Mittels solcher Austausch-Aufgaben ist es möglich, die Kontrolle ganz in die Hand der

Lerner zu legen, welche so selbst feststellen, ob die Vektordarstellung verstanden wurde oder ob Fehler unterlaufen sind. Dieses Vorgehen ermöglicht es den SuS, die Betreuer des Moduls von Beginn an als Unterstützer und nicht als Lehrer wahrzunehmen.

Direkt im Anschluss an diese erste praktische Aufgabe erfolgt der Übergang zur Transformation (erst einmal nur die Skalierung) solcher Objekte. Dies geschieht indem diese zunächst per Hand auf dem Nagelbrett vergrößert werden, dann die neuen Vektoren abgelesen werden und zum Schluss die entsprechende Übergangsmatrix schrittweise zusammengesetzt wird. Auf diese Art und Weise können die Lerner die Transformationsmatrix selbstständig erarbeiten, was wesentlich motivierender und somit auch effektiver ist, als nur gegebene Informationen aufzunehmen (vgl. [Gu07]).

Phase 1: Koordinatentransformation

Nachdem die Vergrößerung von Grafiken per Hand auf dem Steckbrett erlernt wurde, werden nun auch die anderen Arten der Transformation (Rotation und Translation) mithilfe eines zu diesem Zweck entwickelten Applets (siehe dazu Abbildung 2) erforscht. Auch hier verfolgt das Modul wieder die Leitidee des selbstentdeckenden Lernens, indem die Aufgabenstellungen (siehe grauer Block links) absichtlich sehr kurz gehalten sind und das Applet viel Raum zum Ausprobieren bietet. Die Lerner haben die Möglichkeit, die Matrixeinträge (oben links) zu verändern und die resultierenden Effekte direkt in der Animationsfläche (rechts) zu sehen oder auch die Grafik direkt per Maus zu verändern, woraufhin die Einträge der Matrix automatisch angepasst werden. Neben dem freien Ausprobieren bietet das Applet insgesamt 13 kleine Aufgaben zur Transformation 2- und 3-dimensionaler Objekte. Für jede Aufgabe gibt es einen Fortschrittsbalken, der den SuS sichtbar macht, wie weit sie noch vom Ziel entfernt sind, und nach Abgabe der Lösung ein Feedback (siehe Box). Um auf die zu Beginn erwähnte große Heterogenität von Informatik-Kursen zu reagieren, steht es den SuS frei, wie viele der vorgegebenen Aufgaben sie absolvieren bzw. inwieweit sie frei ausprobieren möchten. Da jedoch nach dieser Phase ein bestimmtes Mindestwissen bei allen SuS vorhanden sein muss, um das Modul sinnvoll fortzusetzen, folgt eine explizite Sicherungsphase.

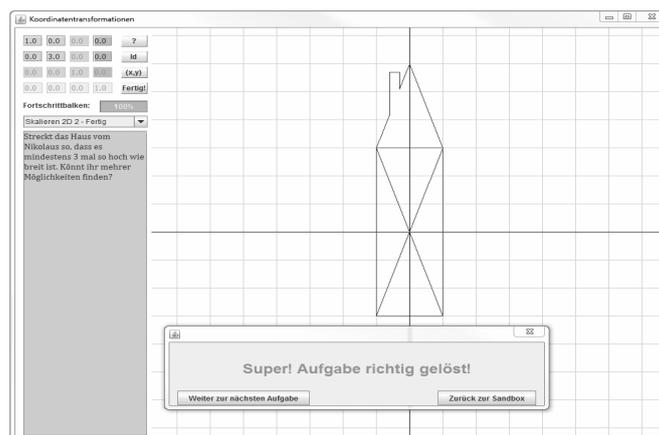


Abbildung 2: Applet "Koordinatentransformation" mit Feedback

Sicherung der 1. Phase

In dieser Phase wird von den Lernern zunächst in Partnerarbeit ein Sicherungsblatt bearbeitet. Dabei ist zu einer bestimmten textuell beschriebenen Transformation die entsprechende Matrix aufzustellen und auch aus einer gegebenen Matrix die entsprechenden Transformationen abzuleiten. In der letzten Aufgabe werden die Matrizen für Skalierung, Rotation und Translation noch einmal allgemein festgehalten (die Musterlösung ist in Abbildung 3 dargestellt). Dies fördert zum einen das abstrakte Denken der SuS und soll zum anderen als Nachschlagemöglichkeit bei der späteren Programmierung dienen. Anschließend werden die Ergebnisse im Plenum besprochen, um sicherzustellen, dass die Grundlagen von allen SuS korrekt verstanden wurden.

$$S = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad R = \begin{pmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad T = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Abbildung 3: Allgemeine Transformationsmatrizen für Skalierung (S), Rotation um die z-Achse (R) und Translation (T)

Einstieg in Phase 2: Präsentation des Programms „Shadermanager“

Nachdem in der ersten Phase die Grundlagen der Objektdarstellung und Transformation inklusive der entsprechenden mathematischen Methoden gelegt wurden, geht es in Phase 2 um die Programmierung dieser Effekte. Speziell für diese Phase wurde das Programm „Shadermanager“ nach dem Vorbild des Shader-Maker (siehe Abschnitt 2) entwickelt, welches ohne tiefergehendes Wissen die Erstellung einfacher Grafikprogramme ermöglicht. Der Shadermanager stellt bewusst nur eine Teilmenge der Funktionalitäten des Shader-Makers zur Verfügung, punktet dabei allerdings durch ein aufgeräumtes Design, einfache Bedienung, umfangreiche Hilfe und verbesserte Fehlerausgaben, so dass die Lernenden ohne ablenkende Einstellungsmöglichkeiten ihre ersten eigenen Programme im Shadermanager implementieren können. Zur Unterstützung erfolgt an dieser Stelle eine kurze Präsentation der Software und der nötigen Grundlagen in der Programmiersprache OpenGL Shading Language durch die Betreuer.

Phase 2: Animation durch Koordinatentransformationen

In dieser Phase geht es nun darum, die im Einstieg erlernten Kenntnisse selbstständig an einfachen Grafikprogrammen zu erproben und zu festigen. Wie bereits oben erwähnt, sind die Arbeitsaufträge in dieser Phase recht konkret und es gibt einige Tipps und Hilfen, um allen SuS einen erfolgreichen Start in die Shaderprogrammierung zu ermöglichen. So erlernen die SuS schrittweise die Gestaltung erster Animationen von Objekten. Dabei finden immer wieder Verweise zur (Schul-)Mathematik statt, damit die Lerner das neuerworbene Wissen in bekannte Strukturen integrieren können. Auch in dieser ersten Phase der Shaderprogrammierung können die SuS sich bereits kreativ ausleben, da nicht festgelegt ist, welches Objekt animiert werden soll und wie die Animation exakt aussehen soll. Dies ermöglicht es auf das individuelle Lerntempo der

SuS einzugehen, da die Animationen beliebig weit ausgebaut und verfeinert werden können.

Phase 3: RGB-Farbdarstellung

Um den SuS auch die freie Farbgestaltung ihrer Objekte zu ermöglichen, erfolgt an dieser Stelle ein Exkurs in die RGB-Farbdarstellung. Dazu steht eine kleine elektronische Schaltung aus einer 9V-Blockbatterie, einer RGB-LED mit Vorwiderstand und drei Potentiometern (siehe Abbildung 4) zur Verfügung. Die Einstellung der Potentiometer (von 0=aus bis 1=voll) bildet dabei eine Analogie zu den RGB-Farbwerten (jeweils 0-255). Somit erlernen die Teilnehmer allein durch Experimentieren wie die Rot-, Grün- und Blauanteile zu einer Mischfarbe zusammen spielen.

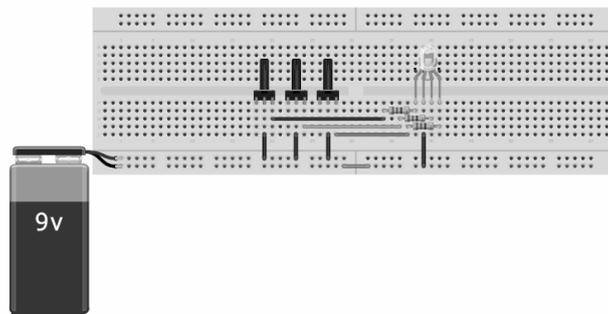


Abbildung 4: Elektrische Schaltung zur RGB-Farbdarstellung

Phase 4: Comiclook

Diese Phase unterstreicht besonders gut die Verknüpfung der Mathematik und Informatik im Themenbereich Computergrafik, da zur Erzeugung des sogenannten Comiclooks (siehe Abbildung 5) die Berechnung des Winkels zwischen der Blickrichtung und der Oberfläche des Objekts (Skalarprodukt der Richtungs- bzw. Normalenvektoren) notwendig ist, um dann den entsprechenden Teil des Objekts je nach Lage zur Blickrichtung unterschiedlich dunkel einzufärben. Dies kann selbstverständlich nicht nur in Grautönen, sondern auch in beliebigen Farbverläufen umgesetzt werden, indem die SuS das in der vorhergehenden Phase erworbene Wissen über die RGB-Farbdarstellung einsetzen.

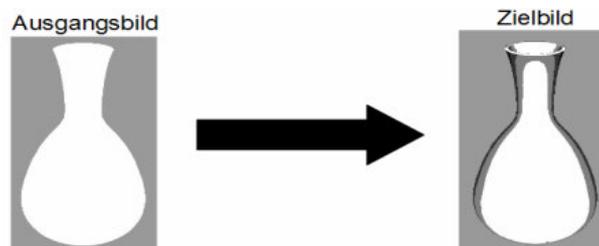


Abbildung 5: Veränderung durch Comiclook

Phase 5: Lokale Beleuchtung

In der letzten obligatorischen Phase geht es um einen besonderen Beleuchtungseffekt, die Phong-Beleuchtung. Dieses Thema eignet sich besonders gut als Abschluss, da das erlernte Wissen über die RGB-Farbdarstellung aus Phase 3 und die Winkelberechnung aus Phase 4 noch einmal in neuem Zusammenhang vertieft wird. Hier stellt das Aufgabenblatt lediglich einen groben Leitfaden für eine mögliche Herangehensweise zur Verfügung, der gleichzeitig den schwächeren Lernern noch etwas Sicherheit gibt und den schnelleren SuS viele Freiheiten aufzeigt. Auch hier wird noch einmal die Notwendigkeit mathematischer Grundlagen für die informatische Umsetzung deutlich.

Exkurs: 3D-Stereo

Je nach Schülergruppe kann dieser Exkurs von der gesamten Gruppe oder auch von einzelnen besonders interessierten und motivierten SuS als Einblick in das Themengebiet 3D-Effekte angeschlossen werden. In dieser Phase erzielen die SuS einen 3D-Tiefeneffekt, indem sie ein und dasselbe Objekt in zwei verschiedenen Farben (Rot und Türkis) leicht versetzt auf dem Bildschirm ausgeben. Dieses wird anschließend für die Person vor dem Bildschirm mit Hilfe einer Rot-Türkis-Brille als 3-dimensionales Objekt sichtbar.

5 Fazit

Mit dem Modul „Wie kommt das Bild auf den Bildschirm? - Einstieg in die Computergrafik“ kann das InfoSphere – Schülerlabor Informatik zukünftig allen Studieninteressierten einen realistischen, zielgruppengerechten und experimentellen Einblick in die Welt der Informatik und damit verbunden der Verknüpfung von Mathematik und Informatik, geben.

Während der Entwicklung wurde das Modul mit zwei Lerngruppen (einem Informatik-Kurs der Einführungsphase und einzelnen interessierten Schülerinnen beim Schnupperstudium der RWTH Aachen) testweise durchgeführt. Diese Durchführungen dienten der formativen Evaluation, Verbesserung der Arbeitsmaterialien und letzten Anpassungen an die Zielgruppe.

Weitere Erfahrungen werden wir am 1. März 2013 bei der Durchführung mit einem Informatik-Leistungskurs und während der Schüleruni 2013 in den Sommerferien sammeln. Durch die in diesem Rahmen stattfindende quantitative Evaluation wird es uns möglich sein, auch den Erfolg des Moduls zu messen.

Literaturverzeichnis

- [Be06] Bell, T.: Schülervorstellungen und Lernen von Physik. Forschendes Lernen, Kiel, 2006.
- [BHS12] Bergner, N.; Holz, J.; Schroeder, U.: Concept of an Extracurricular Learning Environment for Computer Science. In (Knobelsdorf, M.; Romeike, R. Hrsg.): PRE-PROCEEDINGS 7th Workshop in Primary and Secondary Computing Education WiPSCE 2012, 2012; S. 26–33.
- [ELK10] Engeser, S.; Limbert, N.; Kehr, H.: Studienwahl Informatik. Abschlussbericht zur Untersuchung, München, 2010.
- [En04] Engeln, K.: Schülerlabors. authentische, aktivierende Lernumgebungen als Möglichkeit, Interesse an Naturwissenschaften und Technik zu wecken. Dissertation. Logos-Verl, Kiel, 2004.
- [Eu05] Euler, M.: Schülerinnen und Schüler als Forscher: Informelles Lernen im Schülerlabor. In Naturwissenschaften im Unterricht. Physik, 2005, 16; S. 4–12.
- [GP08] Guderian, P.; Priemer, B.: Interessenförderung durch Schülerlaborbesuche, 2008.
- [Gr94] Griffin, J.: Learning to learn in informal science settings. In Research in Science Education, 1994, 24; S. 121–128.
- [Gu07] Guderian, P.: Wirksamkeitsanalyse außerschulischer Lernorte. Der Einfluss mehrmaliger Besuche eines Schülerlabors auf die Entwicklung des Interesses an Physik. Dissertation, Berlin, 2007.
- [He11] Hechenberger, S.: Das informatische Weltbild von Studienanfängern der Informatik. Empirische Untersuchung unter Absolventen der letzten G9-Jahrgänge in Bayern, München, 2011.
- [MN11] MNU Deutscher Verein zur Förderung des mathematischen und naturwissenschaftlichen Unterrichts e.V. Hrsg.: 033.068 MNU BS 2011 X3.indd. Aufgaben im naturwissenschaftlichen Unterricht. Lernen – Prüfen - Diagnostizieren. Verlag Klaus Seeberger, Neuss, 2011.
- [MW06] Maass, S.; Wiesner, H.: Programmieren, Mathe und ein bisschen Hardware ... Wen lockt dies Bild der Informatik? In Informatik-Spektrum, 2006, 29; S. 125–132.
- [Re10] Reher, J.: Dossier über die Informatik in der allgemeinbildenden Schule für die Bundesländer Nordrhein-Westfalen und Niedersachsen. Dossier, Potsdam, 2010.
- [Un12] Universität Bremen: Computer Graphics - Teaching. http://cgvr.cs.uni-bremen.de/teaching/shader_maker/index.shtml, 06.02.2013.

Konzeption eines Informatik-Moduls für ein Schülerlabor

Jonas Neugebauer, Dieter Engbring

FG Didaktik der Informatik
Universität Paderborn
Fürstenallee 11
33102 Paderborn

jonas.neugebauer@upb.de
didier@upb.de

Abstract: In diesem Beitrag wird der Aufbau eines Informatik-Moduls für Schülerinnen und Schüler der Sekundarstufe II im Rahmen eines MINT-Schülerlabors beschrieben und evaluiert. Zur Gestaltung des Moduls wurden LEGO Mindstorms NXT Bausteine eingesetzt. Aus den Erfahrungen ergeben sich Hinweise auf die Gestaltung und Bewertung ähnlicher Angebote, zur Nutzung von Mindstorms NXT im regulären Informatikunterricht und weitergehende Forschungsfragen, die insbesondere Unterrichtsprojekte in den Blick nehmen.

1 Einleitung

Es gibt inzwischen eine ganze Reihe von Schülerlaboren. Ihr Aufbau wird u.a. in der Absicht gefördert, mehr Auszubildende und Studierende für den Bereich der sog. MINT-Fächer zu akquirieren¹, zu denen auch die Informatik gehört. Interesse für diese wichtigen, aber häufig unterrepräsentierten Fächer zu wecken, ist aber nur ein Ziel. Darüber hinaus sollen unterrichtliche Themen in einer Art und Weise vertieft werden, die in den Schulen so nicht oder nur schwer umsetzbar sind. Insbesondere mit dieser zweiten Intention ist das in diesem Beitrag dargestellte Informatik-Modul für Schülerinnen und Schüler (im folgenden nur noch Schüler) entstanden. Dort wurden Aspekte einer komplexen Infrastruktur – eines Container-Hafen – basierend auf LEGO Mindstorms NXT modelliert, die in dieser Komplexität an Schulen wohl nicht umsetzbar sind.

In diesem Beitrag werden wir, ausgehend von der Positionierung des Schülerlabors, die grundlegenden Überlegungen zur Etablierung eines Informatikmoduls für die Sekundarstufe II (SII) darstellen, bevor die Bestandteile des Moduls beschrieben werden. Erste Evaluationsergebnisse können vorgelegt und daran anknüpfend ein Ausblick auf zukünftige Weiterentwicklungen und Forschungsfragen geworfen werden. Die Forschungsfragen betreffen nicht nur die Einrichtung von Schülerlaboren als außerschulischem Lernort bzw. als Bestandteil der Lehrer-(fort-)bildung, sondern auch den Einsatz von LEGO Mindstorms im regulären Informatikunterricht, sowie die (Weiter-)Entwicklung von fachdidaktischen Konzepten, die ein hohes Maß an Eigenverantwortung der Schüler und an projektorientiertes Vorgehen enthalten, wie dies z.B. in der systemorientierten Didaktik der Informatik der Fall ist. Deren zentraler

¹Vgl. z.B. http://eldiss.uni-kiel.de/macau/receive/dissertation_diss_00003669

Bestandteil – das Informatik Lernlabor – stand auch Pate bei der Konzeption unseres Informatik-Moduls.

2 Zur Einrichtung coolMINT.Paderborn

Das Schülerlabor coolMINT.Paderborn ist in den Jahren 2009 und 2010 als eine gemeinschaftliche Einrichtung des Heinz Nixdorf MuseumsForum (HNF) und der Universität Paderborn geplant und implementiert worden. Wie am Namen schon zu erkennen, sind neben der Informatik auch andere Fächer aus dem MINT-Bereich vertreten. Gefördert wurde der Aufbau durch die Initiative „Zukunft durch Innovation“ (zdi)², die in Nordrhein-Westfalen mehrere solcher Einrichtungen finanziert hat. Im Verlauf der Antragsstellung wurden verschiedene, auch fächerübergreifende Module und verschiedene Rahmenbedingungen für den Modulzuschnitt diskutiert. Von fächerübergreifenden Modulen wurde letztlich zugunsten der einfacheren Buchung Abstand genommen: Schulunterricht ist in Fächern organisiert und durch fachspezifische Module können die notwendigen Lernvoraussetzungen von den buchenden Lehrern besser abgeschätzt werden. Weiter wurde festgelegt, dass alle Module in ein Zeitraster von drei Zeitstunden eingepasst werden sollen. Bei der Planung des im Folgenden beschriebenen Informatik-Moduls für die SII musste diese nicht ganz einfach einzuhaltende Rahmenbedingung auch beachtet werden. Wie schon erwähnt, sollen darüber hinaus alle Fachmodule ihre Inhalte auf eine Weise aufbereiten, die so im alltäglichen Unterricht nicht umsetzbar ist und damit ein attraktives Angebot mit echtem Mehrwert liefert. Andererseits sollte das Modul aber inhaltlich zumindest an den Richtlinien und Vorgaben zum Zentralabitur anknüpfen.

3 Grundlegende Überlegungen zum Informatik-Modul

Von den oben genannten Anforderungen an ein Modul im coolMINT Schülerlabor stand bei dem nun im Folgenden beschriebenen Informatik-Modul für die SII die Attraktivität an erster Stelle. Unsere Erfahrungen mit mechatronischen Systemen haben gezeigt, dass mit diesen ansprechende Umgebungen für Schüler aufgebaut werden können. Durch die Zeitvorgabe von drei Stunden, und das Ziel, über die Möglichkeiten des Schulunterrichts hinauszugehen, konnte bei der Konstruktion eines solchen Systems nicht bei Null begonnen werden. Daher wurde beschlossen, ein funktionsfähiges System bereitzustellen. Da dies ein solides Grundwissen informatischer Inhalte voraussetzt, wurden als Zielgruppe vor allem fortgeschrittene Schüler in den Blick genommen, die sich in der Informatik vertiefen möchten.

Grundlage für das Modul waren Vorarbeiten aus einem universitären fachdidaktischen Seminar mit dem Titel „Informatik Lernlabor“ (ILL). In diesem Seminar befassen sich die Studierenden aus der Perspektive der Lehrenden wie auch der Lernenden mit dem zentralen Bestandteil der systemorientierten Didaktik [Ma03]. Dabei geht es als didaktische Großform eines informatischen Unterrichtsprojektes darum, ein bestehendes Informatiksystem zu analysieren (dekonstruieren), zu rekonstruieren und dann

²Zukunft durch Innovation NRW: <http://www.zukunft-durch-innovation.de/>

weiterzuentwickeln. Das Informatiksystem spiegelt als Modell einen Ausschnitt der Wirklichkeit wieder und ist damit in einen gesellschaftlichen Kontext eingebettet.

In dem ILL-Seminar waren im Jahr 2009 mit Hilfe von LEGO Mindstorms auf der Basis von NXT-Bausteinen Artefakte und Abläufe eines automatisierten Containerhafens nachgebildet worden. Diese Vorarbeit war die Basis und zugleich Ideengeber für vier Teilmodule mit überschaubaren Aufgabenstellungen (vgl. Tabelle 1), die im Rahmen von Staatsexamensarbeiten weiter ausgearbeitet wurden ([En11], [Ne11], [Th11], [Wö10]).

Name des Teilmoduls	curricularer Bezugspunkt
Verladekran	Datenstrukturen (Stapel, Hash-Tabellen)
Fernsteuerung	Datenstrukturen (Liste)
Datenbanken	Datenbanken, Abfragesprachen
Kommunikation	Kommunikationsprotokolle, ISO/OSI-Referenzmodell

Tabelle 1: Die implementierten Module

Ziele waren die Konstruktion eines mechatronischen Artefakts und die Entwicklung methodisch wie inhaltlich abwechslungsreicher Aufgaben. Neben den umgesetzten Themen gibt es eine Reihe von weiteren Vorschlägen, deren Realisierung aus unterschiedlichen Gründen bislang nicht vorangetrieben werden konnte. Dazu gehören:

- Positionsbestimmung zur Navigation eines Fahrzeuges in der Ebene z.B: über Kompassensoren
- Navigation eines Fahrzeuges unter Berücksichtigung der aktuellen Position und der Position von Hindernissen
- Entwicklung eines Kollisionsvermeidungskonzepts (Fahrabschnittskontrolle) für autonome Transportfahrzeuge

4 Beschreibung der Teilmodule

Durch die thematische Eingliederung in den Kontext eines automatisierten Containerhafens wurden für die Teilmodule Aspekte aufgegriffen, die für die Einbettung der curricularen Bezugspunkte geeignet erschienen. Zur Umsetzung der Datenstrukturen wurden als Artefakt jeweils Portalkräne gewählt, wie sie charakteristisch für das Erscheinungsbild eines Containerhafens sind, und die mit hoher Wahrscheinlichkeit für die meisten Schüler ein bekanntes Element darstellen. Die beiden anderen Stationen bauen auf den autonomen Transportfahrzeugen auf, die in automatisierten Hafenanlagen Schiffscontainer zwischen Kränen, Lagerplätzen und Verladestationen transportieren. Im folgenden beschreiben wir die Teilmodule im Einzelnen.

4.1 Gemeinsamkeiten der Teilmodule

In den drei Zeitstunden, die den Schülern zur Verfügung stehen, wird vor allem Wert darauf gelegt, dass die Teilnehmer möglichst selbstständig mit den vorhandenen Modellen arbeiten. Daher wird nur zu Beginn eine kurze Einführung in die Arbeitsumgebungen gegeben, um einen Überblick über das vorhandenen Material zu

verschaffen. Der in Java verfasste Programmcode wird in der Entwicklungsumgebung NetBeans³ bearbeitet, die neben den üblichen Funktionen wie Syntaxhervorhebung und Codevervollständigung auch die Integration von Javadoc-APIs erlaubt. Außerdem lassen sich die Programme mit nur einem Mausklick auf die NXT Bausteine übertragen. Neben dem Quelltext steht eine virtuelle Erkundungsumgebung in Form eines Wikis zur Verfügung. In dieser werden zusätzliche Informationen zu den Systemen bereitgestellt, die zum selbstständigen Bearbeiten der Module von Nutzen sind.

Die angebotenen Aufgaben stellen eine Mischung aus eher theoretischen und eher experimentellen Problemstellungen dar. Erstere sollen die Schüler dazu anregen, sich mit den konzeptionellen Schwerpunkten der Teilmodule zu befassen. Die experimentellen Aufgaben dienen der Umsetzung der entwickelten Konzepte und laden zum „herumbasteln“ ein.

4.2 Verladekran: Automatisierte Lagerung von Containern

Für das Teilmodul wurde bei der Umsetzung die Variante eines Verladekrans gewählt, der Container auf festgelegte Lagerplätze verteilen kann. Jeder Lagerplatz hat eine bestimmte Kapazität und kann von dem Hubwagen angesteuert werden. Die Implementierung basiert auf Arrays und die Schüler stoßen bei der Erweiterung der Funktionalität bald auf die Grenzen dieser Datenstruktur. Das Auswählen einer passenderen Alternative (Stapel) und das Rekonstruieren des Systems stellen hier wichtige Ziele dar. Als weitere Aufgabe soll ein RFID-Sensor genutzt werden, um Chips an den Containern auszulesen. Auf diese Weise können den einzelnen Frachtstücken Informationen zugeordnet werden, die innerhalb des Programms in einer Hash-Tabelle abgelegt sind. Die Schüler erweitern eigenständig das Programm und nutzen die vorhandenen Java-APIs zum Zugriff auf Hash-Tabellen.

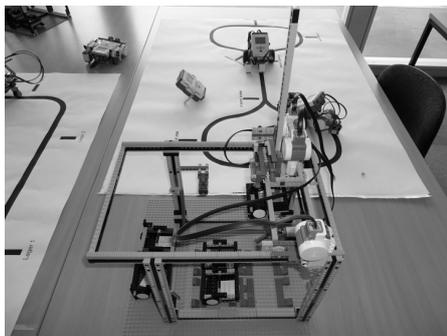


Abbildung 1: Teilmodul "Verladekran"



Abbildung 2: Teilmodul "Fernsteuerung"

³<http://netbeans.org>

4.3 Fernsteuerung: Programmierung einer Notfallfernsteuerung

Ähnlich wie das erste Teilmodul bildet das Artefakt des Moduls Fernsteuerung einen Portalkran nach. Im Unterschied dazu arbeitet dieser Kran nicht vollständig autonom, sondern wird über eine Fernbedienung gesteuert. Diese ist mit Hilfe eines zweiten NXT Bausteins erstellt worden und kann über eine Bluetooth Verbindung Befehle an den Kran schicken. Um die manuelle Steuerung zu erleichtern soll eine Makro-Funktion implementiert werden, mit der es möglich ist, spezielle Ablauffolgen des Krans einzuspeichern. Die Ansätze einer solchen Funktion sind schon vorhanden und müssen von den Schülern ergänzt und vervollständigt werden. Die Makros werden als Positionsangaben der Motoren in einer Liste gespeichert. Deren Implementierung orientiert sich, als Bezug zum Unterricht, vor allem an den Vorgaben für das Zentralabitur 2012 [Mi10]. Die Listen werden in Textdateien gespeichert, wodurch Konzepte wie Ein- und Ausgabestreams genutzt werden müssen.

4.4 Datenbank: Datenbank gesteuerter Transport von Containern

Für die Station zum Thema Datenbanken wurde ein autonomes Fahrzeug entworfen, das einer schwarzen, zyklischen Linie folgt und anhand von Markierungen am Streckenrand seine Position erkennen kann. Dadurch kann das Fahrzeug völlig autonom Aufträge abarbeiten, bei denen zum Beispiel ein Container von einem Lager zu einem Anlieger gebracht werden soll. Die Aufträge werden von einem datenbankgestützten PC-Programm über Bluetooth an das Fahrzeug gesendet. Zunächst können über eine textbasierte Benutzeroberfläche Datenabfragen gestellt und Aufträge an das Fahrzeug gesendet werden. Durch das Erkunden des Systems wird die Aufteilung der Daten in verschiedene Tabellen sowie die Verknüpfung über Primärschlüssel sichtbar. Die Schüler nehmen Änderungen an der Datenbank vor, indem Sie neue Tabellen einfügen und die Benutzerschnittstelle um neue Abfragemöglichkeiten erweitern. Dabei ist es nicht notwendig, den Umgang mit Abfragesprachen zu beherrschen, da dafür eine objektorientierte Abstraktionsschicht vorgegeben wird. Dies bietet den Vorteil, dass sich durch die Typsicherheit von Java der Aufbau von Abfragen schon aus der Methodensignatur der einzelnen Abfrage-Klassen ergibt.

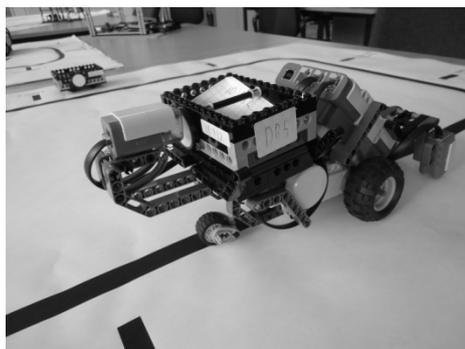


Abbildung 3: Datenbank

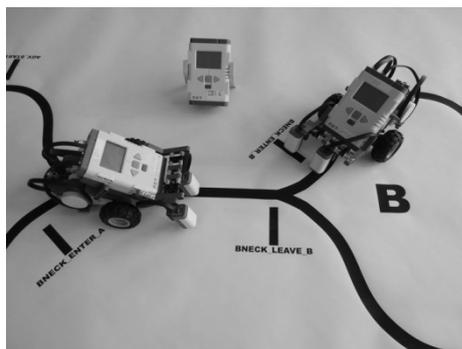


Abbildung 4: Kommunikation

4.5 Kommunikation: Interaktionen autonomer Transportfahrzeuge

Zentrale Themen des vierten Teilmoduls sind Kommunikationsprotokolle und das ISO/OSI-Referenzmodell. Das vorhandene Informatiksystem besteht aus zwei autonomen Transportfahrzeugen, die eine sich kreuzenden Kreisbahn befahren. Um Kollisionen zu vermeiden kommunizieren beide Fahrzeuge mit einem zentralen Server, der für die Sperrung und Freigabe der kritischen Streckenabschnitte zuständig ist. Die Schüler erkunden die Bestandteile von Protokollen anhand der vorhandenen Kommunikation und lernen zwischen den Schichten des ISO/OSI-Modells zu unterscheiden. Dabei wird der Fokus auf die drei obersten Schichten (Anwendung, Präsentation, Sitzung) gelegt. Unter anderem werden Aufgaben zur Differenzierung zwischen verschiedenen Bedeutungen des Begriffs Protokoll und ein „Schichtenpuzzle“, bei dem die Schichten des Referenzmodells in eine sinnvolle Reihenfolge gebracht werden sollen, angeboten. Zum Schluss wird zusätzlich zum vorhandenen Kommunikationsserver ein „Piratenserver“ aktiviert. Dieser versucht durch Störsignale eine Blockade der Strecke oder eine Kollision der Fahrzeuge zu provozieren. Da das Verbindungsprotokoll keine Identifikation der Kommunikationspartner vorsieht, kann der Piratenserver einen kritischen Streckenabschnitt beim Server als frei melden, den er vorher nicht als besetzt markiert hat. Zur Absicherung der Kommunikation müssen die Schüler nun ihr bisher erworbenes Wissen einsetzen und selbstständig eine Lösung des Problems erarbeiten.

5 Evaluation

5.1 Evolution des Moduls

Das Modul „Informatik im Containerhafen“ mit seinen oben beschriebenen vier Teilmodulen hat seit der ersten Konzeption wesentliche Änderungen durchlaufen. Zu verschiedenen Zeitpunkten wurden Testdurchläufe mit anschließender Überarbeitungsphase angesetzt. Nach Fertigstellung der ersten Modulversionen durch Studenten musste zunächst eine Reduktion des, als Obermenge von möglichen Aufgaben angelegten, Umfangs vorgenommen werden. Bei einem Test mit Hilfe eines Kurses der Jahrgangsstufe 12 wurde deutlich, dass der Umfang des Programmcodes zu groß war. In der abschließend durchgeführten informellen Befragung gaben die Teilnehmer an, dass in drei Stunden die Einarbeitung zwar möglich wäre, aber daneben keine sinnvolle Bearbeitung der Aufgaben zu schaffen sei. Diese seien aber ansprechend und zielgruppengerecht gestaltet. Die Arbeit mit der NetBeans Umgebung, die den Umgang mit den umfangreichen Quelltexten erleichtern sollte, stellt eher eine Hürde als eine Hilfe dar, da den Schülern die Funktionen zum Großteil unbekannt waren. Als Konsequenz wurden für die Schüler unwichtige Klassen in externe Bibliotheken ausgelagert, wodurch sich der Umfang des Quelltextes nun auf eine überschaubare Anzahl von Klassen beschränkt. Durch die Möglichkeiten von NetBeans ist es einfach die Bibliothek zusammen mit der Dokumentation direkt in der Entwicklungsumgebung verfügbar zu machen. Damit die Vorteile der Plattform auch genutzt werden, wurde die kurze Einführungsphase zu Beginn der Arbeitszeit ergänzt. Durchführungen mit weiteren Kursen haben diesen veränderten Aufbau des Moduls als sinnvoll bestätigt.

5.2 Lehrerfortbildung im Schülerlabor

Um den Kontakt zur Lehrerschaft herzustellen und das Modul zu bewerben, wurde im Herbst 2012 eine Fortbildung im Rahmen des Moduls Informatik im Hafen durchgeführt. 16 Lehrerinnen und Lehrer hatten Gelegenheit die Module (Artefakte und Aufgaben) zu erkunden und Erfahrungen mit den Systemen zu sammeln. Im Anschluss wurde Feedback zusammengetragen und eine Diskussion über Möglichkeiten zur Verknüpfung von außerschulischem Lernen und Informatikunterricht angeregt. Trotz einiger technischer Probleme konnte ein überwiegend positives Feedback eingeholt werden. Unter Berücksichtigung der besonderen Voraussetzungen, die das Modul an die Kurse stellt, schätzten die anwesenden Lehrkräfte den Schwierigkeitsgrad und den Umfang als angemessen ein. Die Kombination der Themengebiete stieß dagegen auf Kritik. Durch die breite Fächerung der gewählten Themen über die gesamte Qualifikationsphase⁴ fiel es den Anwesenden schwer, den Besuch im Labor auf sinnvolle Weise mit dem Unterricht zu verknüpfen, da die teils erheblichen Unterschiede in den Voraussetzungen als Zeitpunkt für einen Besuch nur das Ende der Q2 zulassen würden. Obwohl die thematische Breite des Moduls aus unserer Sicht einen attraktiven Aufbau darstellt, scheint dies aus Lehrersicht eher nicht der Fall zu sein, und der sinnvollen Einbettung in den Unterricht im Wege zu stehen. Um diesem Problem zu begegnen, wurde vorgeschlagen, die vier Teilmodule in vier vollwertige, thematisch fokussierte Module aufzuteilen, die auch einzeln gebucht werden können. Um dies zu ermöglichen werden die technischen Artefakte derzeit vereinheitlicht und angepasst um vier Kopien jedes Modells verfügbar zu haben, so dass 16 bis 20 Schüler das gleiche Teilmodul bearbeiten können. Dadurch würden aber theoretisch auch beliebige Zusammenstellungen der Teilmodule ermöglicht.

5.3 Befragung der Workshop-Besucher

Zur weiteren Evaluation wurde seit Ende 2012 zum Abschluss der Bearbeitungszeit sowohl an die Schüler, als auch an die betreuenden Lehrkräfte ein kurzer Fragebogen ausgegeben. Die Teilnehmer wurden nach der zuletzt bearbeiteten Aufgabe gefragt und aufgefordert, mit eigenen Worten die Inhalte zu beschreiben. Dann folgten insgesamt 21 Items, die auf einer 4-Stufigen Likert-Skala von „Trifft überhaupt nicht zu“ (1) bis „Trifft voll und ganz zu“ (4) beurteilt werden sollten. Im Itempool befanden sich Fragen zu Schwierigkeit und Umfang der Aufgaben, der Funktionsweise der Modelle, der Einstellung der Befragten zum Modul und dem Schülerlabor, sowie der Einbettung des Besuchs im Unterricht. Den Lehrkräften wurden vor allem Fragen zur Verknüpfung mit dem Informatikunterricht gestellt. Darunter, ob der Besuch im Labor mit größeren Umständen (lange Fahrtzeit, zusätzliche Kosten) verbunden war. Der Fragebogen wurde in Bezug auf eine ähnliche Evaluation in einem Aachener Schülerlabor erstellt [BHS12].

Im Dezember 2012 und Januar 2013 buchten 5 Klassen mit insgesamt 68 Schülern das Modul. Von diesen waren zwei Kurse der Q1 (LK und GK) und drei der Q2 (zwei LKs

⁴In Nordrhein-Westfalen gibt es seit ein paar Jahren ein Nebeneinander von sog. G8 und G9-Abitur, so dass man die Jahrgangsstufen nicht mehr mit 11, 12 und 13 bezeichnet. Stattdessen setzt man die Bezeichnungen Einführungsphase für das erste Jahr und Qualifikationsphase 1 und 2 (Q1, Q2) für die beiden weiteren Jahre.

und ein GK) vertreten. Der Besuch im Schülerlabor war in der Regel mit einem Besuch im MuseumsForum oder einem anderen Programmpunkt verbunden (nur eine Klasse kam ausschließlich für das Modul „Informatik im Hafen“). Dazu kamen teils erhebliche Fahrtzeiten von über drei Stunden. Die Auswertung der Schülerdaten zeigt, dass die Vorerfahrungen mit Lego Mindstorms eher gering waren. Für 58% der Befragten traf die Aussage „Ich hatte schon zuvor Erfahrungen mit Lego Mindstorms“ nicht zu. Nur bei einem Kurs mit 18 Schülern waren Mindstorms zuvor Bestandteil des Unterrichts. Überraschenderweise zeigt der Bearbeitungsfortschritt, dass die Schüler mit Vorerfahrungen eher weniger Aufgaben bearbeiteten, als diejenigen ohne. Während die letzte Gruppe sich mit etwas über 60% der bereitgestellten Probleme befasste, waren es bei der ersten Gruppe nur etwa 53%. Obwohl diese Beobachtung aufgrund der geringen Anzahl an Befragten Schülern nicht als aussagekräftig gewertet werden kann, bietet sie Anlass für zukünftige Untersuchungen. Eine mögliche Erklärung ist, dass viele Gruppen Aufgaben übersprungen haben (vgl. Abschnitt 9), und so die Anzahl der tatsächlich bearbeiteten Aufgaben von der Angabe des Fortschritts abweicht.

Das Ziel, ein komplexes Informatiksystem zu erschaffen, das über die Möglichkeiten des Schulunterrichts hinaus geht, konnte nur zum Teil erfüllt werden. Die Schüler schätzten die Komplexität der beiden Teilmodule Verladekran und Fernsteuerung deutlich höher ein als die der beiden Teilmodule Kommunikation und Datenbank (Item: „Die Modelle waren komplizierter als die Modelle mit denen ich bisher gearbeitet habe.“). Dies ist wenig überraschend, da die Arbeit mit Mindstorms in der Regel mit Fahrzeugen begonnen wird. Die Portalkräne entsprechen hier auf den ersten Blick eher dem Anspruch, neue Erfahrungsfelder zu öffnen. Andererseits stellen auch die Programmpakete zur Steuerung der Fahrzeug-Module hohe Ansprüche an die Schüler. Da in der aktuellen Evaluation die Software nicht berücksichtigt wurde, können hierzu keine Aussagen getroffen werden. Dies wird in Zukunft nachgebessert.

Die Aufgabenstellungen wurden als nicht zu schwer und nicht zu leicht eingeschätzt. Die Aussagen „Die Arbeit mit dem Modul hat mir Spaß gemacht“ und „Ich würde gerne wieder das Schülerlabor besuchen“ wurden deutlich positiv beurteilt. Das Selbe gilt für die eigenständige Arbeitsform im Labor, die 97% der Befragten gefallen hat. Das zeigt sich auch daran, dass 80% der Befragten angaben, bei der Bearbeitung etwas Neues gelernt zu haben. Andererseits scheint sich die neuen Kenntnisse nicht unbedingt auf die zentralen Themen der einzelnen Stationen zu beziehen, denn etwa 60% gaben an, dass ihnen die Inhalte des Moduls schon vor dem Laborbesuch bekannt waren. Die Einschätzungen der Schüler können aber zunächst als positiv gewertet werden, da das Lerndesign des Moduls sie zum Arbeiten motiviert und mit den Thematiken in Berührung bringt. Die nächsten Schritte sollten nun darauf zielen zu prüfen, ob durch die Arbeit am Informatikmodul ein Beitrag zur Entwicklung der informatischen Kompetenzen der Schüler geleistet werden kann. Um dies zu evaluieren müssen gezielte Kompetenztests mit den Schülern durchgeführt werden. Ansätze dazu können zum Beispiel im MoKoM Projekt [Li12] gefunden werden, in dem unter anderem ein Messinstrument für Kompetenzen im Bereich der Systemmodellierung entwickelt wurde.

Das gewünschte Ziel einer Einbettung in den Schulunterricht ist mit dem Informatikmodul derzeit nur bedingt erreicht. Nur eine Lehrkraft gab an, den Besuch im Schüler-

labor als Abschluss einer Unterrichtsreihe zu nutzen. Eine mögliche Erklärung dafür lässt sich im Feedback der Lehrerfortbildung (s.o.) erkennen. Die thematische Vielfalt des Moduls steht dem direkten Einbezug in den Verlauf einer Unterrichtsreihe im Weg. 43% der Schüler konnten dennoch Anknüpfungspunkte zum vorangegangenen Unterricht finden und gaben an, dass der Besuch gut zum Unterricht der letzten Wochen gepasst habe. Die Auswertung zeigt aber auch, dass zur Verknüpfung des Schülerlabor Moduls mit dem Unterricht noch weitere Änderungen notwendig sein werden.

5.4 Weitere Beobachtungen

Die Aufgabe der Moderatoren ist es, jeweils auf Nachfrage, oder falls sie auf Schwierigkeiten aufmerksam werden, Hilfestellungen zu geben. Da einige Schülergruppen nur sehr zögerlich dazu bereit sind, von sich aus Hilfe einzufordern, haben die Moderatoren den Auftrag, regelmäßig die Fortschritte der Gruppen zu überprüfen. Dabei konnten die beobachten, dass die Schüler konzentriert an ihren Teilmodulen arbeiteten und die Aufgaben im wesentlichen selbstständig lösten.

Besonders auffällig bei den Schülergruppen war, dass sie sich nicht an den vorgegebenen Arbeitsablauf gehalten haben. Vor allem wurden Aufgaben ohne Einbezug der Modelle oder ohne direkte Implementierung häufig übersprungen bzw. nicht sorgfältig bearbeitet. Gerade die Aufgaben, die zu einer tieferen konzeptionellen Verständnis beitragen sollen, wurden nur selten bearbeitet. Beispielsweise wurde das „Schichtenpuzzle“ des Kommunikations-Teilmoduls, die als unplugged Aufgabe gestaltet ist, nur von einer der Schülergruppen bearbeitet. Durch das Auslassen dieser Aufgaben stellt sich die Frage, in wie weit die Schüler den beabsichtigten Einblick in die vorgestellten Thematiken bekommen. Zur Beantwortung dieser Frage wird es notwendig sein Vor- und Nachtests in den Schulen durchzuführen.

6 Fazit und Ausblick

Mit diesem Modul sollen Schüler angesprochen werden, die für sich selbst schon einen Schwerpunkt in der Informatik gesetzt haben. Sie sollen ihre Erfahrungen und Kompetenzen im Fach Informatik vertiefen. Dies gelingt dahingehend, dass die Schüler selbstständig entlang der Aufgabenstellungen an den Informatiksystemen arbeiten. Allerdings werden dabei häufig gerade diejenigen Aufgaben übersprungen, die der konzeptionellen Vertiefung bzw. Festigung und nicht der Weiterentwicklung dienen. Anhand der Beobachtungen kann exemplarisch gezeigt werden, ob und wie Unterricht mit komplexeren, in einem Kontext stehenden Systemen möglich ist und ob dabei Richtlinien konform und gemäß der Vorgaben gearbeitet werden kann. Es gilt zukünftig herauszufinden, wo bei der Umsetzung solcher didaktischer Ansätze Grenzen verlaufen und wie die Schüler dazu animiert werden können, auch bei eher konzeptionellen Aufgaben konzentriert und engagiert zu arbeiten. Hierzu mag auch die Kooperation mit einem Gymnasium im Rahmen eines Informatik-Projektkurses dienen, der mit dem Ziel startet, zunächst ein Teilmodul aus Schülersicht weiterzuentwickeln. Durch weitere Fortbildungen sollen daneben auch Lehrerperspektiven berücksichtigt werden und hoffentlich weitere Schulklassen für detailliertere Evaluationen gewonnen werden.

Aus technischer Sicht hat der Aufbau der Module die Grenzen der LEGO Mindstorms Education-Sets – des Basiskastens für den Schuleinsatz – aufgezeigt. Für kurze Sequenzen, die einen Einstieg in die Programmierung bieten, lassen sich die mitgelieferten Bauteile gut einsetzen. Allerdings ist das Einsteiger-Modell sehr schnell ausgereizt, will man nicht hauptsächlich basteln, sondern Konzepte vermitteln. Mechanik und Sensorik komplexerer mechatronischer Systeme lassen die erforderliche Präzision vermissen, was nicht nur die Lust derer sinken lässt, die nicht mit intrinsischer Motivation am Informatikunterricht teilnehmen. Wollte man LEGO Mindstorms nicht nur im AG-Bereich, z.B. für Roboter-Wettkämpfe, sondern tatsächlich im Informatikunterricht nutzen, müsste man nicht nur diese Probleme beseitigen, sondern auch für mindestens drei Jahrgangsstufen in Kursstärke Material vorhalten. Bei üblichen Kursgrößen von 20 bis 30 Schülern entsteht nicht nur finanzieller Aufwand sondern auch Bedarf an Lagerkapazitäten, da die Artefakte aufgebaut bleiben und vor (un-)absichtlicher Zerlegung geschützt werden müssen. Für Roboter- oder Informatik-AGs mit hoch motivierten Schülern ist hierin aber weiterhin ein attraktives Betätigungsfeld zu finden.

7 Literaturverzeichnis

- [BHS12] Bergner, N., Holz, J., Schroeder, U.: InfoSphere: An Extracurricular Learning Environment for Computer Science. In: Knobelsdorf, M., Romeike, R. (Hrsg.) Proceedings of 7th Workshop in Primary and Secondary Computing Education (WiPSCE 2012). Hamburg. (2012) (in press)
- [En11] Ende, G.: Didaktische und softwaretechnische Entwicklung einer Lernstation zur Übertragung mittels RFID-Technik und Farbsensoren im Rahmen des ZDI-Schülerlabors mit den Schwerpunkten Leistungsvergleich und der Anbindung an Datenbanken. Staatsarbeit, Universität Paderborn. (2011)
- [Li13] Linck, B., Ohrndorf, L., Nelles, W., Neugebauer, J., Magenheim, J., Schaper, N., Schubert, S., Stechert, P.: Competence model for informatics modelling and system comprehension. 4th IEEE Global Engineering Education Conference (IEEE EDUCON 2013). Berlin. (2013) (in press)
- [Ma03] Magenheim, J.: Informatik-Lernlabor. Systemorientierte Informatik in der Praxis. In: Hubwieser, P. (Hrsg.): Informatische Fachkonzepte im Unterricht, Proceedings der Infos 2003, 10.GI-Fachtagung Informatik und Schule, 17.-19. September in Garching bei München, S. 13-31
- [Mi10] Ministerium für Schulen und Weiterbildung Wissenschaft und Forschung des Landes Nordrhein-Westfalen. Vorgaben zu den unterrichtlichen Voraussetzungen für die schriftlichen Prüfungen im Abitur in der gymnasialen Oberstufe im Jahr 2012, 2010
- [Ne11] Neugebauer, J.: Didaktische und softwaretechnische Entwicklungen für eine Station im Rahmen des ZDI-Schülerlabors mit dem Schwerpunkt „Kommunikationsprotokolle autonomer LEGO-Mindstorms Fahrzeuge“. Staatsarbeit, Universität Paderborn. (2011)
- [Th11] Thiessen, A.: Didaktische und softwaretechnische Entwicklung einer Lernstation „Fernsteuerung“ im Rahmen des ZDI-Schülerlabors mit dem Schwerpunkt der Aufzeichnung und Bearbeitung von Bewegungsabläufen des Verladekrans durch Verwendung der ADT Liste. Staatsarbeit, Universität Paderborn. (2011).
- [Wö10] Wöstefeld, S.: Didaktische und softwaretechnische Entwicklung einer Lernstation im Rahmen des Schülerlabors mit dem Themenschwerpunkt „Containerverladung mit Sensortechnik“. Staatsarbeit, Universität Paderborn. (2010)

Vermittlung von Konzepten der Informationstechnik in Oberstufen der beruflichen Bildung unter Verwendung mobiler Roboter als Lernträger

Lars Otten

Bildungszentrum für Technik und
Gestaltung Oldenburg
Straßburger Straße 2
26123 Oldenburg

Zentrum für Didaktik der Technik
Leibniz Universität Hannover
Appelstraße 9A
30167 Hannover

lars.otten@bztg-oldenburg.de

Abstract: „RoboTea“ (Robotics for Technical Education Teaching Concept) ist ein Verfahren, das den Einsatz mobiler Roboter in Oberstufen technischer Fachrichtungen der beruflichen Bildung ermöglicht und der Vermittlung grundlegender Konzepte der Informationstechnik dient. Lehrkräften werden Lernmodule bereitgestellt, die sich an zu vermittelnden fachlichen Kompetenzen orientieren und u.a. Hinweise zu möglichen Unterrichtsabläufen und Unterrichtsmaterialien beinhalten. Ein wesentliches RoboTea-Merkmal ist der unterrichtliche Einsatz des als Lernträger dienenden mobilen Roboters „myBot“ (my Robot), der von den Schülerinnen und Schülern individuell konstruiert wird und den jede Schülerin und jeder Schüler zur zeit- und ortsungebundenen Nutzung besitzt.

1 Einführung

Ein wesentlicher Bestandteil der beruflichen Bildung technischer Fachrichtungen ist die Vermittlung von Kenntnissen aus dem Bereich der Informationstechnik, wobei der Praxisbezug ein wesentliches Element der Ausbildung darstellt. Im Gegensatz zur traditionellen dualen Ausbildung, bei der ein betrieblicher Dualpartner für die praktische Ausbildung zur Verfügung steht, erfolgt die Ausbildung innerhalb von Oberstufen beruflicher Bildung mit Ausnahme eventueller betrieblicher Praktika rein schulisch. Zu diesen Schulformen gehören insbesondere Berufliche Gymnasien sowie Berufsoberschulen und Fachoberschulen, die auf ein Studium an einer Fachhochschule bzw. Universität vorbereiten. In diesen Schulformen

- fehlt der in der dualen Ausbildung vorhandene betriebliche Dualpartner, so dass der damit verbundene betriebliche Betroffenheitsaspekt und die daraus resultierende Motivationsquelle nicht vorhanden sind,
- finden sich häufig in einem Klassenverband Schülerinnen und Schüler mit unterschiedlicher beruflicher Spezialisierung. So verfügen z.B. Schülerinnen und Schüler der Berufsoberschule Technik zwar bereits über eine abgeschlossene berufliche Ausbildung im technischen Bereich, diese Berufe können jedoch aus unterschiedlichen Fachrichtungen wie z.B. der Elektrotechnik, der Informationstechnik oder der Metalltechnik stammen.

RoboTea soll

- Möglichkeiten aufzeigen, wie die Vermittlung grundlegender Konzepte der Informationstechnik unter Einsatz mobiler Roboter in Oberstufen berufsbildender Schulen technischer Fachrichtungen erfolgen kann,
- hinsichtlich unterschiedlicher Schwierigkeitsgrade und unterschiedlicher fachlicher Schwerpunkte aus dem Bereich Informationstechnik anpassbar sein,
- dem konstruktivistischen Ansatz folgend eine hohe Eigenaktivität der Schülerinnen und Schüler beinhalten,
- eine hohe Motivation der Schülerinnen und Schüler bewirken, sich mit der Thematik mobiler Roboter und so mit den Inhalten der Informationstechnik zu befassen.

Die Idee des Einsatzes von Computern und darauf aufbauend von Robotern als Lernträger in der Schule stammt aus dem Ende der 70er Jahre und basiert auf dem Mathematiker und Psychologen Seymour Papert, dem Entwickler der Programmiersprache LOGO [PA80]. Insbesondere für den allgemeinbildenden Bereich existiert inzwischen eine Vielzahl von Untersuchungen bzgl. des Unterrichtseinsatzes der Robotik. So konnte dort unter anderem nachgewiesen werden, dass der Einsatz von Robotik die Schülerinnen und Schüler motiviert, sich mit dieser Thematik zu beschäftigen ([PE04], [RU04], [SC05]) und hierbei das Interesse an Technik fördern kann [SC05]. Erfahrungen im berufsbildenden Bereich zeigen ebenfalls eine hohe Motivation bei Einsatz von Robotern als Lernträger [MO12]. Es zeigt sich jedoch auch, dass der Robotereinsatz in fachfremden, d.h. nichttechnischen Berufen von den Auszubildenden als wenig hilfreich für die Ausbildung angesehen wird [ST08].

Unterrichtliche Robotikprojekte weisen häufig nur einen allgemeinen Bezug zum Lehrplan auf (z.B. Förderung des Technikinteresses [SC05]) oder beziehen sich lediglich auf spezielle Themengebiete (z.B. Automatisierungstechnik [MO12]). Hilfreich wäre jedoch auch ein Konzept, das zeigt, wie die Robotik in den täglichen informationstechnischen beruflichen Unterricht zur Förderung spezifischer fachlicher Kompetenzen eingebunden werden kann. Mit RoboTea wird ein Konzept bereitgestellt, das diese Anforderung unterstützt und die Vermittlung einer hohen Bandbreite fachlicher Inhalte der Informationstechnik derart ermöglicht, dass die Anwendung des Konzeptes nicht nur punktuell, sondern während der gesamten Ausbildungsdauer erfolgen kann.

2 Das RoboTea-Konzept

Bei RoboTea handelt es sich um ein Konzept, das Hilfestellung geben soll, wie mobile Roboter in Oberstufen berufsbildender Schulen technischer Fachrichtungen während der gesamten Ausbildung zur Vermittlung informationstechnischer Inhalte eingesetzt werden können. Diese Hilfestellung umfasst

- Anleitungen und Hinweise zur Planung und Durchführung des robotikbasierten Unterrichts,
- die Bereitstellung exemplarischer Unterrichtseinheiten zur Vermittlung wesentlicher Inhalte der Informationstechnik z.B. zu den Themen Aufbau und

Funktionsweise von Rechnern, Programmierung, Grundlagen der Elektrotechnik und Steuerungstechnik,

- Hinweise zu den zu verwendenden mobilen Robotern.

Kern des RoboTea-Konzeptes sind Module, die entsprechend der roboterbasierten Vermittlung fachlichen Kompetenzen aufbereitet sind. Hierbei besteht die Möglichkeit der Modulanpassung durch die Lehrkraft bzgl. geplanter Lehr-/Lernziele. Da der Unterricht an berufsbildenden Schulen nach dem didaktischen Konzept der Handlungsorientierung durchzuführen ist [NK11], beinhaltet jedes Modul Lernsituationen, die handlungs- und berufsorientierte Aufgabenstellungen und Lernarrangements beinhalten. Hierbei wird die von S. Papert definierte und auf dem Konstruktivismus basierende Didaktik des Konstruktivismus berücksichtigt, bei der das Lernen als förderlich angesehen wird, wenn Elemente durch den Lernenden individuell hergestellt, d.h. konstruiert werden [PA94]. Auf den Einsatz der Robotik bezogen bedeutet dieses, dass der Bau von Roboterkomponenten sowohl hard- als auch softwaremäßig nicht fest vorgegeben ist, sondern Freiheitsgrade beinhaltet, die individuelle Lösungen der Lernenden zulassen.

Bei der roboterbasierten Vermittlung informationstechnischer Inhalte kann der Robotikeinsatz den zentralen Unterrichtsgegenstand darstellen, einen vertiefenden Charakter aufweisen oder auch nur unterrichtsbegleitend erfolgen. Entsprechend der von der Lehrkraft durchzuführenden Unterrichtsplanung berücksichtigen die RoboTea-Module diese Einsatzarten. Der mobile Roboter besitzt dabei jeweils die Funktion eines Lernträgers.

Das wesentliche Merkmal des RoboTea-Konzeptes ist die Verwendung mobiler Roboter als Lernträger mit denen die Lernenden z.B. eigene Experimente durchführen können. Untersuchungen zeigen, dass die Nutzung eines eigenen Experimentiersets, auf das man zeit- und ortsunabhängig Zugriff hat, eine Erhöhung der fachliche Kompetenz bewirkt gegenüber der Nutzung von Experimentieranordnungen, die lediglich in der Schulungseinrichtung zur Verfügung stehen [GR06]. Übertragen auf den Einsatz mobiler Roboter bedeutet dieses, dass der Roboter nicht nur in der Schule, sondern auch zuhause z.B. zur Durchführung von Programmierübungen verwendet werden sollte. Jeder Lernende erhält daher einen eigenen Roboter (Bezeichnung „myBot“ für „my Robot“), der in seinem Besitz ist und während der gesamten Ausbildungszeit verwendet wird.

2.1 RoboTea-Lernmodule

Das RoboTea-Konzept ist so gestaltet, dass es bzgl. der zu vermittelnden Inhalte und der Dauer der Inhaltsvermittlung anpassbar und erweiterbar ist. Ebenso ist adaptierbar, ob der Einsatz mobiler Roboter als zentraler Unterrichtsgegenstand, unterrichtsbegleitend oder zur Vertiefung informationstechnischer Inhalte erfolgen soll.

Die Flexibilität des Unterrichtskonzeptes wird durch Lernmodule erreicht, die in Abhängigkeit der jeweils geforderten Lehr-/Lernziele eingesetzt werden. RoboTea-Lernmodule beschreiben Lernziele, Eingangsvoraussetzungen, Lerninhalte, Lernsituationen, mögliche Unterrichtsabläufe und Übungen. Sie sind ein Hilfsmittel für

die Unterrichtsplanung und stellen Unterrichtsmaterialien wie z.B. Leittexte und Informationen zur Verfügung.

Für die Unterrichtsplanung können mittels der Lernmodule Lernpfade festgelegt werden, die den jeweiligen unterrichtlichen Schwerpunkten entsprechen. Abbildung 1 zeigt eine Übersicht vorhandener Lernmodule, bei denen mögliche Lernpfade mittels Pfeilen gekennzeichnet sind. Die gestrichelten Pfeile geben exemplarisch einen Lernpfad an, bei denen die Module I1 (Einführung in die Informationstechnik), E (Grundlagen der Elektrotechnik), P1 (Imperative Programmierung) und S (Einführung in die Steuerungstechnik) der Reihe nach bearbeitet werden.

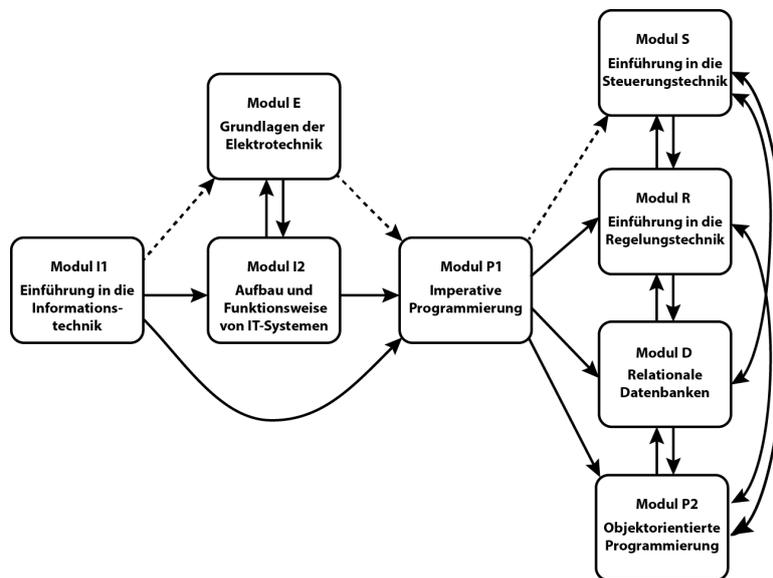


Abbildung 1: RoboTea-Module mit Lernpfaden

Inhaltlich-didaktisch-methodische Leitlinien der RoboTea-Lernmodule:

- Jedes RoboTea-Modul wird einem informationstechnischem Themenbereich zugeordnet, innerhalb dessen die Vermittlung fachlicher Kompetenzen erfolgt. Derzeit wird zwischen Modulen folgender Themenbereiche unterschieden: Informationstechnik (I), Elektrotechnik (E), Programmierung (P), Steuerungstechnik (S), Regelungstechnik (R), Datenbanken (D)
- Ein durch ein Modul repräsentiertes Themengebiet kann in weitere Module unterteilt werden, wobei die Kennzeichnung der Modulthemen dann mit aufsteigender Nummerierung erfolgt. So wird das Modul „Programmierung“ (P) z.B. unterteilt in die Module P1 „Imperativen Programmierung“ und das darauf aufbauende Modul P2 „Objektorientierte Programmierung“.
- Die Lernsituation beinhaltet eine robotikbasierte und produktorientierte Problemstellung, die unterschiedliche Lösungsansätze ermöglicht.

- Die Lernmodule sind so konzipiert, dass deren sequentielle Abarbeitung einen Lernpfad darstellt. Die Lehrkraft wählt einen Lernpfad und damit die zu vermittelnden fachlichen Schwerpunkte.
- Handlungsorientierter Unterricht ist aufgrund des Aufbaus ein induktiv ausgerichtetes Unterrichtskonzept, bei dem ausgehend von einer konkreten Lernsituation auf allgemeine Grundsätze geschlossen wird. Da es sich bei dem mobilen Roboter jedoch um ein bereits recht komplexes technisches System handelt, wird in den Modulen neben der induktiven Vorgehensweise eine Vermittlung angestrebt, die inhaltlich vom „Groben“ zum „Feinen“ führt. Ausgehend von „Black Boxes“, bei denen zunächst die Ein- und Ausgabe betrachtet und somit ein Überblickswissen vermittelt wird, erfolgt anschließend der Übergang zu dem inneren Aufbau und der Funktionsweise dieser Black Boxes.
- Jedes RoboTea-Modul enthält Übungsaufgaben in zwei Schwierigkeitsstufen (grundlegend, erhöht). Entscheidet sich die Lehrkraft, das RoboTea-Modul nicht als zentralen Unterrichtsgegenstand unter Verwendung der enthaltenen Lernsituation zu verwenden, sondern das Modul unterrichtsbegleitend einzusetzen, so werden lediglich die grundlegenden robotikbasierten Übungsaufgaben bearbeitet. Bei Verwendung des RoboTea-Moduls als vertiefender Einsatz werden die Übungsaufgaben der erhöhten Schwierigkeitsstufe bearbeitet.

2.2 Allgemeiner RoboTea-Unterrichtsverlauf

Im Sinne des geforderten handlungsorientierten Unterrichts erfolgt der Unterrichtsverlauf innerhalb eines RoboTea-Moduls im Rahmen einer vollständigen Handlung, in der die Phasen „Informieren“, „Planen/Entscheiden“, „Umsetzen“, „Kontrollieren/Auswerten“ solange nacheinander durchlaufen werden, bis die Problemlösung zufriedenstellend erfolgt ist. Die zeitlichen Rahmen der einzelnen Phasen sind abhängig von der zu vermittelnden fachlichen Kompetenz und der jeweiligen Schülerklientel und sind somit nicht einheitlich. Jedes RoboTea-Modul mündet in der Erstellung eines hard- und/oder softwaremäßigen Produktes.

Bei gewünschter Durchführung eines RoboTea-Unterrichts sind von der Lehrkraft im Rahmen der Unterrichtsplanung zunächst folgende Elemente festzulegen:

- a) Lernpfad (RoboTea-Module): Welche fachlichen Kompetenzen sollen vermittelt werden?
- b) Einsatzart: Soll der Robotereinsatz als zentraler Unterrichtsgegenstand, unterrichtsbegleitend oder zwecks Vertiefung bereits vorhandener Kenntnisse erfolgen?
- c) Falls der Robotikeinsatz als zentraler Unterrichtsgegenstand erfolgt: Entspricht die in dem RoboTea-Modul enthaltene Lernsituation den eigenen Vorstellungen? Sind hier ggf. Anpassungen vorzunehmen?
- d) Zeitrahmen: Entsprechen die in den RoboTea-Modulen enthaltenen Zeitrachwerte der unterrichtlichen Situation z.B. bzgl. der eigenen Schülerklientel?

Nach Vorstellung der robotikbasierten Lernsituation durch die Lehrkraft, informieren sich die Lernenden in Gruppenarbeit zunächst über bislang unbekannt Sachverhalte der

Problemstellung. In der sich anschließenden Planungsphase wird die Problemlösung geplant, d.h. es wird festgelegt, welche Schritte notwendig sind, um das robotikbasierte Problem zu lösen. Nach der Planung erfolgt die Umsetzung der geplanten Lösungsschritte, so dass hier die eigentliche Roboterprogrammierung oder –konstruktion erfolgt. Wurde das Problem nicht zufriedenstellend gelöst, informieren sich die Lernenden erneut, planen wiederum die Problemlösung, usw. Bei zufriedenstellender Problemlösung präsentieren die Schülergruppen ihre Ergebnisse anschließend im Plenum. Da es sich bei den Lernsituationen grundsätzlich um konkrete Problemstellungen handelt, erfolgt im Anschluss an die Präsentationsphase noch eine Zusammenfassung und Verallgemeinerung der Problemstellung in Form eines Lehrer-Schüler-Gesprächs, so dass ein Transfer der konkreten Problemstellung auf ähnliche Problemstellungen für die Lernenden erleichtert wird. Abschließend erfolgt die Bewertung der Schülerleistungen durch die Lehrkraft.

3 Das mobile Robotersystem „myBot“

In robotikbasierten Unterrichtseinheiten werden häufig Fertigroboter wie z.B. FESTO-Robotino [MO12] oder Roboterbausätze wie z.B. LEGO-Mindstorms [SC05] verwendet. Fertigroboter wie z.B. FESTO-Robotino [FE12], NI LabVIEW Robotics Starter Kit [NI12] oder Volksbot [FR12] mit Preisen zwischen 1600 Euro und 6500 Euro¹ sind für viele berufsbildende Schulen nicht finanzierbar. Die RoboTea-Anforderung, dass jeder Lernende einen eigenen Roboter erhält, mit dem nicht nur in der Schule, sondern auch zuhause gearbeitet werden kann, ist damit nicht realisierbar. Gegen die Verwendung von Fertigrobotern spricht ebenfalls die im Sinne des von S. Papert definierten Konstruktivismus fehlende Möglichkeit der individuellen Konstruktion und die damit verbundene niedrigere Identifikation und Motivation bzgl. der Beschäftigung mit diesem System.

Roboterbausätze wie z.B. LEGO-Mindstorms [LE13] oder Fischertechnik-ROBO [FI13] erfüllen zwar die Anforderung, eigene Konstruktionen vornehmen zu können, sie liegen mit Preisen zwischen 200 Euro und 300 Euro¹ jedoch ebenfalls in Bereichen, bei denen die Anschaffung eines Systems pro Schüler kaum finanzierbar ist. Diese Roboterbaukästen besitzen zudem den Nachteil, dass Technik aufgrund der Anwender ab dem Grundschulalter weitestgehend „versteckt“ wird. Eigene Erweiterungen sind häufig nur schwer möglich und benötigen Spezialbauteile (z.B. LEGO-Spezialstecker). In der beruflichen Bildung technischer Fachrichtungen sollte Technik jedoch erkennbar und messtechnisch leicht verifizierbar sein. Ebenso sollten Änderungen und eigene Erweiterungen bzgl. der bestehenden Technik durchführbar sein. Diese Anforderungen erfüllt z.B. die im Hochschulbereich u.a. in der Lehre eingesetzte mobile Robotikplattform MoRob [WA04], die modular aufgebaut ist und eine Vielzahl von Hard- und Software-Schnittstellen für eigene Erweiterungen bereitstellt. Da mit diesem mobilen System jedoch auch komplexe und schwere Elemente transportierbar sind, ist es entsprechend groß und mit Kosten von einigen tausend Euro pro System kaum finanzierbar. In Anlehnung an MoRob wurde innerhalb von RoboTea daher ein

¹ Stand März 2013

Robotiksystem entwickelt, das ebenso wie MoRob modular aufgebaut ist und über eine Vielzahl von Schnittstellen für eigene Erweiterungen verfügt, jedoch klein (17cm x 11cm x 11cm, LxBxH) und kostengünstig (ca. 80 Euro bis 150 Euro) ist. Dieses System ist von den Lernenden zeit- und ortsunabhängig einsetzbar, so dass es sowohl während des Unterrichts als auch zuhause z.B. zur Durchführung von Programmierübungen nutzbar ist. Jeder Lernende erhält daher einen eigenen mobilen Roboter, der von ihnen käuflich erworben wird. Die Lernenden sind hierbei für die individuelle Gestaltung ihres Roboters sowie die sachgerechte Umgangsweise mit dem System verantwortlich. Das mobile Robotersystem erhält daher auch die Bezeichnung „myBot“ als Abkürzung für „my Robot“ – der eigene Roboter der Schülerinnen und Schüler.

Das myBot-System besitzt folgende Eigenschaften:

- **Mechanischer Aufbau:**
Das myBot-System ist kein fester Bausatz, sondern besteht aus Komponenten, die von den Lernenden individuell zusammengesetzt, angepasst und erweitert werden können.
- **Programmierung:**
Zur Einführung in die Programmierung eignet sich die grafische Programmierumgebung „Scratch“ [MI13]. Sie stellt ebenfalls einen geeigneten Übergang zur Programmierung mittels höherer Programmiersprachen dar [LE07]. Programmierbefehle werden bei Scratch als Puzzleteile dargestellt, die nur dann zusammenpassen, wenn sie syntaktisch korrekt angeordnet sind. Syntaxfehler sind bei der Programmierung somit ausgeschlossen. Aufgrund der Eignung der Scratch-Programmierung für Programmierneinsteiger, wurde diese Programmiermöglichkeit auch für das myBot-System entwickelt und bereitgestellt. Zusätzlich ist das myBot-System mittels gängiger Hochsprachen wie z.B. C++ oder C# programmierbar. Ein typisches Einsatzgebiet der Programmierung in technischen Berufen ist die Automatisierung technischer Anlagen. Daher wurde eine Applikation entwickelt, die die myBot-Programmierung zusätzlich mittels simulierter verbindungsprogrammierter Steuerungen ermöglicht. Hiermit kann nicht nur das myBot-System selbst gesteuert werden, sondern es ist auch eine Steuerung von Modellen technischer Anlagen möglich, die an das myBot-System angeschlossen sind.
- **Steuereinheit:**
Kernelement des myBot-Systems ist eine Steuereinheit, die mittels eines modular aufgebauten Mikrocontrollersystems vom Typ „.NET Gadgeteer“ [MC12] realisiert wird. Das .NET-Gadgeteer-Mikrocontrollersystem enthält standardisierte Steckplätze, über die sich eine Vielzahl von Komponenten wie Sensoren und Aktoren ohne die Notwendigkeit des Lötens anschließen lassen. Schülerinnen und Schüler ohne Vorkenntnisse können mit Hilfe des .Net-Gadgeteer-Systems schnell eigene Roboter-Prototypen zusammenstecken. Fortgeschrittene Schülerinnen und Schüler können hingegen eigene Komponenten wie z.B. die Motorsteuerung selbst entwickeln und über das System betreiben.



Abbildung 2: Variante eines myBot-Systems

4 Bisherige Untersuchungsergebnisse

Anhand von 114 Schülerinnen und Schülern (Alter zwischen 16 und 24 Jahren) eines Beruflichen Gymnasiums Technik mit Schwerpunkt Informationstechnik (Jahrgangsstufe 11 und 12) sowie einer Berufsoberschule Technik (Jahrgangsstufe 13) wird derzeit untersucht, ob die Anwendung des RoboTea-Konzeptes neben der Förderung der Motivation, sich mit den Inhalten der Informationstechnik zu beschäftigen, zu einer signifikanten Förderung fachlicher Kompetenzen in diesem Bereich führt.

Bisherige Ergebnisse:

- a) *Führt die Anwendung des RoboTea-Konzeptes zu einer Erhöhung der fachlichen Kompetenz im Bereich der Informationstechnik?*

Es können bislang Aussagen bzgl. der Bereiche Programmierung und Steuerungstechnik getroffen werden:

Die Programmierung des myBot-Systems mittels der auf Scratch basierenden Programmierumgebung führt bei Programmierneulingen schnell zu Erfolgserlebnissen, so waren 90% (103 Personen) der Lernenden nach vier Unterrichtsstunden (Modul P1) in der Lage, ihnen unbekannte imperativ arbeitende Programme, die Wiederholungs- und Entscheidungsstrukturen beinhalten, korrekt zu analysieren.

Die Bearbeitung des Lernmoduls der Steuerungstechnik (S) wird derzeit von 14 Schülergruppen (3 bis 4 Personen pro Gruppe) durchgeführt. Es wurde die myBot-Motorsteuerung entwickelt und umgesetzt, so dass die hierzu notwendigen elektrotechnischen Inhalte korrekt erarbeitet wurden. Alle Schülergruppen haben korrekt funktionierende Steuerungen zur Implementierung der Roboterbewegung entwickelt und umgesetzt, so dass die hierzu notwendigen fachlichen Kompetenzen erworben wurden.

- b) *Wirkt die Anwendung des RoboTea-Konzeptes motivierend auf Schülerinnen und Schüler, sich mit Inhalten der Informationstechnik zu beschäftigen?*

75% (86 Personen) der Lernenden zeigte ein hohes Interesse an der Arbeit mit dem myBot-System, das sich u.a. in der Formulierung eigener myBot-Problemstellungen äußerte. 10% (11 Personen) der Lernenden zeigte ein sehr hohes Interesse, indem sie sich außerhalb des Unterrichts selbstständig in die myBot-Programmierung mittels einer höheren Programmiersprache eingearbeitet haben. Völliges Desinteresse war bei keinem Lernenden identifizierbar. 45% (51 Personen) der Lernenden haben in ihrer Freizeit eigene Anpassungen und Erweiterungen an dem myBot-System vorgenommen. Aufgrund des Interesses am Umgang mit dem myBot-System, wurde eine Robotik-AG gegründet, an der zurzeit 15% (17 Personen) der Lernenden wöchentlich in der 9./10. Stunde und somit in ihrer Freizeit teilnehmen.

Innerhalb des bislang umgesetzten RoboTea-Konzeptes (Module I1, P1 und S) konnten folgende Faktoren zur Förderung der Motivation ermittelt werden:

- Schnelle Erfolge bei Programmierneulingen bzgl. der myBot-Programmierung mittels der auf Scratch basierenden Programmierumgebung.

- Die direkte Umsetzung eigener Programme in physikalische Bewegungen. Dieses betrifft sowohl Bewegungen des myBot-Systems, als auch mittels myBot durchgeführte Steuerungen externer technischer Anlagen.
 - Starke Identifikation der Schülerinnen und Schüler mit dem myBot-System und den dort individuell vorgenommenen Anpassungen und Erweiterungen.
 - Ortsunabhängige Verfügbarkeit des myBot-Systems: Da das myBot-System auch zuhause verfügbar ist, können Ergebnisse dort z.B. den Eltern präsentiert werden. Zudem können eigene Anpassungen und Erweiterungen am myBot-System außerhalb der Schule durchgeführt werden.
- c) *Bietet das RoboTea-Konzept eine Hilfestellung bei der Planung und Durchführung informationstechnischen Unterrichts?*
- Rückmeldungen der beteiligten Lehrkräfte zeigen, dass die bereitgestellten Unterrichtshinweise und –materialien eine gute Unterstützung für die Vermittlung darstellen. Die in den Lernmodulen enthaltenen Lernsituationen und zugehörigen Unterrichtshinweise bilden einen „roten Faden“ bzgl. der Vermittlung und stellen somit eine gute Planungsgrundlage dar. Die Unterrichtsmaterialien sind jedoch auch an einigen Stellen zu überarbeiten, so wurde z.B. die Entwicklung der myBot-Motorsteuerung von den Lehrkräften als zu aufwendig beurteilt.

6 Zusammenfassung und Ausblick

Mit RoboTea wurde ein Verfahren vorgestellt, dass in Oberstufen berufsbildender Schulen technischer Fachrichtungen eingesetzt werden kann und der Vermittlung grundlegender Konzepte der Informationstechnik dient. Anhand von Beispielen aus der Programmierung und der Steuerungstechnik konnte gezeigt werden, dass dieses Verfahren geeignet ist, fachliche Kompetenzen aus dem Bereich der Informationstechnik zu vermitteln. Ein wesentliches RoboTea-Merkmal ist der unterrichtliche Einsatz des als Lernträger dienenden mobilen Roboters myBot, der von den Lernenden individuell konstruiert wird und den jeder von ihnen zur zeit- und ortsungebundenen Nutzung besitzt. Die Schülerinnen und Schüler nutzen das myBot-System daher sowohl in der Schule zur Durchführung des RoboTea-Unterrichts als auch zuhause zur Durchführung von Übungen. Es hat sich gezeigt, dass viele Schülerinnen und Schüler das myBot-System auch in ihrer Freizeit nutzen und Programmierungen vornehmen oder das System anpassen und erweitern, so dass hier eine erhöhte Motivation sichtbar wird.

Zurzeit beschränkt sich die RoboTea-Untersuchung auf die RoboTea-Module Informationstechnik (I1), Programmierung (P1) und Steuerungstechnik (S). Für genauere Untersuchungsergebnisse sind weitere Lernmodule auszuarbeiten und zu testen. Ebenso sind die bisherigen Lernmodule zu überarbeiten. So hat sich z.B. gezeigt, dass die Konstruktion und Umsetzung der myBot-Motorsteuerung zu zeitaufwendig ist. Für eine nachhaltige Einführung des RoboTea-Unterrichts ist zudem eine Fortbildung für Lehrkräfte durchzuführen, in der sowohl das RoboTea-Konzept verdeutlicht wird, als auch eine Einführung in den Umgang mit dem myBot-System und dessen vielfältigen Möglichkeiten der Programmierung erfolgt.

Literaturverzeichnis

- [FE12] Festo Didactic: Lernsystem Robotino – Forschen und lernen mit Robotern, Festo Didactic, Denkendorf, 2012
URL: <http://www.festo-didactic.com/de-de/lernsysteme/robotino-forschen-und-lernen-mit-robotern>
- [FI13] Fischertechnik, ROBO
URL: <http://www.fischertechnik.de/home/produkte/computing.aspx>
- [FR10] Bredenfeld, A.; Leimbach, T.; Theidig, G.; Trella, S.; Breuer, T.: Roberta - Grundlagen, Roberta - Reihe Band 1, Fraunhofer IRB-Verlag, Stuttgart, 2010
- [FR12] Fraunhofer Institut, VolksBot, URL: <http://www.volksbot.de/>
- [GR06] Grimheden, M., Hanson, M.: The Lab in Your Pocket – A modular approach to experimental learning in Mechatronics, in *Mechatronics Engineering Education*, Doctoral Thesis M. Grimheden, S.261-273, KTH School of Industrial Engineering and Management, Stockholm, 2006
- [LE07] Leitner, H. H.; Malan, D. J.: Scratch for Budding Computer Scientists, Proceeding SIGCSE'07, S. 223-227, ACM, New York, USA, 2007
- [LG13] LEGO, Mindstorms, URL: <http://mindstorms.lego.com/en-us/Default.aspx>
- [MC12] Hammil, K.; Hodges, S.; Miller, C.; Scott, J.; Villar, N.: .NET Gadgeteer: A Platform for Custom Devices, Pervasive 2012, LNCS 7319, S. 216-233, Springer, Heidelberg, 2012
- [MI13] Lifelog Kindergarten Group, MIT Media Lab: Project Scratch
URL: <http://scratch.mit.edu>
- [MO12] Motz, F.; Riedl, A.: Mobile Robotik im Berufsschulunterricht, in *Die berufsbildende Schule*, Heft 7/8 2012, S. 235-239, dbb Verlag GmbH, Berlin, 2012
- [NI12] National Instruments, NI LabVIEW Robotics Starter Kit für Ausbildung und Lehre,
URL: <http://sine.ni.com/nips/cds/view/p/lang/de/nid/208018>
- [NK11] Niedersächsisches Kultusministerium: Ergänzende Bestimmungen für das berufsbildende Schulwesen (EB-BbS) 2011, Abschnitt 2.7,
URL: http://www.mk.niedersachsen.de/download/61881/Ergaenzende_Bestimmungen_fuer_das_berufsbildende_Schulwesen_2011_-_Lesefassung_.pdf
- [PA80] Papert, S.: Mindstorms: children, computers and powerful ideas, Basic Books Inc., New York, 1980
- [PA94] Papert, S.: Revolution des Lebens, Verlag Heinz Heise, Hannover, 1994
- [PE04] Petre, M.; Price, B.: Using Robotics to Motivate 'Back Door' Learning. In *Education and Information Technologies*, Volume 9, S.147-158, Springer Netherlands, 2004
- [RU04] Ruiz-del-Solar, J.: Robotics Courses for Children as a Motivation Tool. In *IEEE Transactions on Education*, Volume 47, Nr. 4, Santiago, Chile, 2004
- [SC05] Schecker, H.; Schelhowe, H.: Wissenschaftliche Begleitung des Projekts ROBERTA – Mädchen erobern Roboter – Abschlussbericht, Universität Bremen, 2005
URL: http://dimeb.informatik.uni-bremen.de/documents/Wiss.Begl.Abschlussb_Oktober_2005.pdf
- [ST08] Struwe, U.: Abschlussbericht ProfiBot, Fraunhofer - Institut Intelligente Analyse- und Informationssysteme, Kompetenzzentrum Technik-Diversity-Chancengleichheit e.V., Sankt Augustin, Bielefeld, 2008
- [WA04] Wagner, B.; Hohmann, P.; Gerecke, U.; Brennecke, C.: Technical Framework for Robot Platforms in Education, International Conference on Engineering Education and Research, iCEER'04, Olomouc, Czech Republic, 2004

Physical Computing im Informatikunterricht

Mareen Przybylla, Ralf Romeike

Institut für Informatik / Didaktik der Informatik
Universität Potsdam
August-Bebel-Str. 89
14482 Potsdam
przybyll@cs.uni-potsdam.de
romeike@cs.uni-potsdam.de

Abstract: Die Steuerung physischer Systeme hat in Form von Robotik bereits einen festen Platz in der informatischen Bildung eingenommen. So wie interaktive Computersysteme nun eine zunehmende Bedeutung in der Gesellschaft erlangen, ermöglicht die Entwicklung einfach programmierbarer Mikrocontroller eine neue Form kreativer Gestaltung von Informatiksystemen, die als Physical Computing bezeichnet wird und die Erstellung interaktiver Objekte und Installationen zum Ziel hat. Im Beitrag werden Ansätze des Physical Computing im Informatikunterricht diskutiert, Anforderungen an Hardware und Unterrichtsgestaltung formuliert und erste praktische Erfahrungen mit der prototypischen Gestaltungs-, Programmier- und Lernumgebung *My Interactive Garden* dargestellt.

1 Einleitung

Innerhalb der letzten Jahrzehnte erlangten interaktive Computersysteme eine immer größer werdende Bedeutung in der Gesellschaft. Sie sind mittlerweile derart in den Alltag aller Menschen integriert, dass sie vielerorts als intelligente Helfer das Leben erleichtern und bereichern. Dieser Trend spiegelt sich auch in der informatischen Bildung wider und hat – bisher vor allem in Form von Robotik – Einzug in die Schulen erhalten. Für Schüler stellt es offenbar ein besonderes Erfolgserlebnis dar, echte physische Objekte steuern können, was in der Lerntheorie des Konstruktivismus reflektiert wird [PH91]. Mit der Idee des Physical Computing, dem künstlerischen Entwickeln interaktiver Objekte, eröffnen sich auch für den Informatikunterricht neue Ansätze konstruktivistischen Lernens. Im vorliegenden Artikel erfolgt eine Diskussion verschiedener Trends und Möglichkeiten zur Programmierung physischer Objekte im Informatikunterricht. Die von den Autoren entwickelte Gestaltungs-, Programmier- und Lernumgebung „My Interactive Garden“ hat zum Ziel, einen attraktiven Zugang zum Physical Computing in der Schule zu ermöglichen, sowie dessen Stärken und Schwächen zu untersuchen. Nach der Diskussion von Physical Computing als Zugang zur Informatik werden *My Interactive Garden* sowie erste Erfahrungen im Unterrichtseinsatz vorgestellt.

2 Interaktive Systeme

Nachdem lange die Entwicklung von Software den gestalterischen Teil im Informatikunterricht bestimmte, wird mit Mindstorms, PicoBoards, PicoCrickets u. Ä. zunehmend die virtuelle mit der realen Welt in Verbindung gebracht, da der haptische Zugang als wichtig erachtet wird. Im Sinne des konstruktionistischen Lernansatzes erschien es wichtig, Kindern nicht nur das Erschaffen virtueller Objekte zu ermöglichen, sondern programmierbarer Systeme, mit denen in der realen Welt interagiert werden kann. Dies führte zur Entwicklung von Lego Mindstorms. In zahlreichen Unterrichtserfahrungen wird beschrieben, wie mit LEGO Mindstorms meist Fahrzeugroboter und industrielle Anwendungen, wie z. B. Hochregallager, nachgebaut werden (z. B. in [Wa05], [We08]). Kritisiert werden Mindstorms' beschränkte Möglichkeiten zur kreativen Entfaltung, was sich u. a. in einer mangelnden Beteiligung von Mädchen äußert (vgl. [Re07]).

Die technischen Entwicklungen der letzten Jahre haben dazu geführt, dass Computer nicht mehr nur als Maschinen gesehen werden, die Anweisungen erhalten und umsetzen, sondern als interaktive *eingebettete* Medien. Wie auch Roboter kennzeichnen sich eingebettete Systeme dadurch, dass Hard- und Software in ein Gesamtsystem integriert sind. Eingebettete Systeme erfüllen immer einen speziellen Zweck und unterscheiden sich von Robotern vor allem dadurch, dass sie Maschinen oder Geräte nicht auf den industriellen Bereich beschränken und auch den mechanischen Aspekt im Bereich der Akteure nicht betonen.

Um Schülern die Gelegenheit zu bieten, bekannte eingebettete Systeme ihrer täglichen Umgebung genauer zu verstehen, bietet es sich an, diese als Unterrichtsgegenstand zu thematisieren. Oft ist Schülern gar nicht bewusst, wo Informatik in ihrem Alltag überall eine Rolle spielt: beispielsweise in Waschmaschinen, im Auto oder Flugzeug, in der Medizin, in Digitalkameras, Navigationsgeräten, DVD-Playern oder Klimaanlage. Dies geht damit einher, dass Informatik von Schülern häufig als ein Unterrichtsfach wahrgenommen wird, welches abstrakte und realitätsferne Themen behandelt und daher nur für eine spezielle Klientel informatisch talentierter Schüler geeignet ist [Kn11]. Ein konkretes Beispiel zur Thematisierung eines eingebetteten Systems liefern Pelz und Arnhold [PA13] mit der Steuerung einer Modell-Waschmaschine. Einen Unterrichtsversuch mit der Implementierung eingebetteter Systeme in einem Puppenhaus beschreibt Strecker [St11]. Bei beiden Projekten wird der Vorteil der haptischen Erfahrbarkeit als positiv bewertet. Während sich diese Versuche vor allem mit der Nachahmung existierender eingebetteter Systeme beschäftigen, betont *Physical Computing* eine stärkere Einbeziehung von Aspekten aus Kunst und Design und eröffnet damit ein größeres Spektrum kreativer Möglichkeiten.

3 Physical Computing

„Physical Computing uses electronics to prototype new materials for designers and artists. It involves the design of interactive objects that can communicate with humans using sensors and actuators controlled by a behavior implemented as software running inside a microcontroller (a small computer on a single chip)“ [Ba11].

Physical Computing hat in den letzten Jahren verstärkt an Bedeutung gewonnen. Vor allem Künstler und Designer verwenden programmierbare Hardware, um interaktive physische Systeme zu kreieren, welche die analoge Welt wahrnehmen und auf sie reagieren. Physical Computing kann als kreative Tätigkeit verstanden werden, bei der interaktive Objekte und Installationen unter Zuhilfenahme von Bastel-, Kunst- oder Designmaterial hergestellt werden. Physical Computing überträgt die traditionell virtuellen kreativen Möglichkeiten der Informatik durch Einbeziehung von Aspekten aus Kunst und Design in die reale Welt und kann das Fach vielfältiger und damit attraktiver erscheinen lassen. Im Informatikunterricht wird Physical Computing bisher mit Sensor- und Aktorenboards, mit vorgefertigten Sensoren und Aktoren, sowie mit Mikrocontrollerboards durchgeführt.

3.1 Interaktive Objekte mit Sensor- und Aktorenboards

Sensorboards sind einfache Leiterplatten, die mit verschiedenen Sensoren ausgestattet sind und mit einem Computer verbunden werden. So gibt es beispielsweise das PicoBoard, welches mit Klang- und Lichtsensoren, Widerstandssensorik, einem Schalter und einem Schieberegler versehen ist und mit Scratch interagiert [PIC10]. Dadurch ist es möglich, in Scratch real existierende Werte der Umgebung als Eingaben zu benutzen. Nachteilig für die Verwendung in interaktiven Objekten ist jedoch, dass Sensorboards keine Ausgänge besitzen und daher ausschließlich als Eingabegeräte einsetzbar sind. Das SenseBoard der Open University [RS10] behebt diesen Mangel: es verfügt über zusätzliche Ausgänge. Eigens für ihre Studenten wurde als Open-Source-Projekt hierfür die auf Scratch basierende Software „Sense“ entwickelt. Somit konnte Ubiquitous Computing – und damit die Entwicklung eingebetteter Systeme –im universitären Bereich auch für Anfänger auf dem Gebiet zugänglich gemacht werden [Ma12]. Aus dem schulischen Bereich gibt es bisher jedoch keine Erfahrungsberichte. Aufgrund der Einfachheit der Handhabung rechnen wir mit einer erfolgreichen Verbreitung auch im Schuleinsatz, sobald das SenseBoard oder vergleichbare Produkte erhältlich sind.

3.2 Interaktive Objekte mit vorgefertigten Sensoren und Aktoren

Die Trennung von Aktoren und Sensoren von einem Board ermöglicht eine bessere Realisierung interaktiver Objekte im Sinne des Physical Computing. PicoCricket [PIC08] ermöglicht die Nutzung vorgefertigter Sensoren und Aktoren und einer „Drag&Drop“-Programmierungsumgebung bereits im Grundschulbereich. Der Ansatz von PicoCricket geht damit wesentlich über das der Robotik-Systeme hinaus: Es sollen Erfinderbaukästen bereitgestellt werden, die Technologie und Kunst vereinen und somit die Kreativität der Benutzer gezielt ansprechen. Dies entspricht genau der Idee des Physical Computing. Es können Lampen, Motoren und Sensoren ergänzt und somit verschiedenartigste interaktive Projekte realisiert werden, die den persönlichen Interessen der Schüler entsprechen. Die Schüler müssen nur noch die einzelnen Bauteile zusammenstecken und dann in der simpel gehaltenen Umgebung programmieren, um ihre ersten eigenen interaktive Objekte zu erstellen. Von positiven Erfahrungen bei der Erstellung interaktiver Objekte mit PicoCricket, insbesondere zur Interessengewinnung und kreativem handlungsorientierten Lernen, berichten u. a. [Gu10, MKL10, Ru08].

Erste Erfahrungen im Informatikunterricht beschreiben Robotik mit PicoCrickets als “spannenden, interessanten und kreativen Zugang zur Informatik” [RR11].¹

3.3 Interaktive Objekte mit Mikrocontrollerboards

Die vielfältigsten Möglichkeiten ergeben sich aus der Nutzung von Mikrocontrollerboards, wie z. B. Arduino [Ba11], welche auch den Ursprung des Physical Computing darstellen: Eine sehr große Anzahl und Bandbreite an Sensoren und Aktoren kann genutzt werden. Baumann [Bau11] berichtet über den Einsatz von Arduino im Informatikunterricht, in welchem Schüler u. a. eine Ampelsteuerung nachbauen. Verwendet werden ein Arduino-Mikrocontroller und handelsübliche Bauteile. Programmiert wird in der zugehörigen Arduino-Programmierungsumgebung in einer an C/C++ angelehnten Programmiersprache. Als nachteilig ergibt sich die Notwendigkeit der Arbeit mit Steckbrettern und losen Litzen, so dass die entstehenden Projekte mitunter schnell unübersichtlich werden oder ein nur sehr abstraktes Modell des zu realisierenden Gegenstandes darstellen (Abb. 1). Hinzu kommt, dass die feinen Drähte leicht brechen oder aus den Buchsen rutschen können, was zur Folge hat, dass häufig Kontakte geprüft werden müssen. Speziell bei komplexeren Aufbauten kann dies einige Zeit in Anspruch nehmen, von der informatischen Ideen ablenken und schnell frustrierend wirken. Größere Experimente verlangen zudem einiges elektrotechnisches Knowhow.

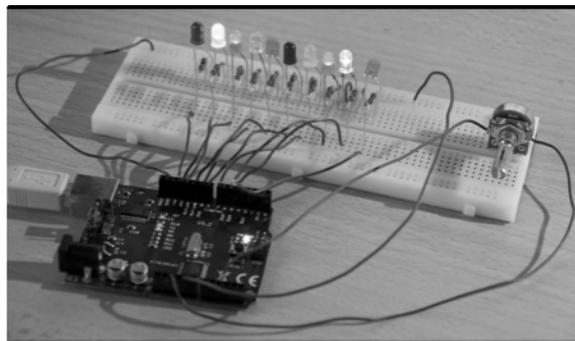


Abbildung 1: Arbeit mit Mikrocontrollerboard, Steckbrett und Litzen

4 Physical Computing als *informatisches Töpfern*

Physical Computing ermöglicht, dass im Informatikunterricht konkrete, physische Produkte der realen Welt entstehen, die der Vorstellungskraft der Lernenden entstammen. Konstruktionistisches Lernen wird damit auf eine haptisch erfahrbare Ebene gehoben und somit konkretisiert. Vahrenhold [Va12] ist der Ansicht, dass dem Informatikunterricht ein „Going to Paris“-Effekt fehlt. Während Schülern und Eltern als Ziel des Französischunterrichts die Möglichkeit, sich auf einer Parisreise verständigen zu können,

¹ Der Vertrieb von PicoCrickets wurde allerdings 2012 eingestellt.

offensichtlich ist, fehlt eine solche Zielvorstellung für den Informatikunterricht, jenseits eines besseren Umgangs mit dem Computer. Unter dem Aspekt des *informatischen Töpfers* bezeichnen wir es als ein Ergebnis des Informatikunterrichts, analog zum Herstellen einer Vase im Töpferkurs, ein digitales, interaktives Artefakt herzustellen, welches im konstruktionistischen Sinn untersucht, herumgezeigt und bewundert werden kann. Aus diesem Verständnis heraus erhält Informatikunterricht einen persönlichen Bezug.

4.1 Kompetenzzuwachs beim Physical Computing

Die von der GI empfohlenen Bildungsstandards Informatik für die Sekundarstufe I [GI08] beinhalten Formulierungen von informatikbezogenen Kompetenzen, die jeder Schüler im Laufe seiner schulischen Informatikausbildung der Sekundarstufe I erwerben soll. Die folgende Übersicht zeigt exemplarisch, welche Kompetenzen in ausgewählten Inhalts- und Prozessbereichen besonders gut beim Physical Computing erworben werden können.

Inhaltsbereiche:

- 1) **Information und Daten:** Der Zusammenhang zwischen Daten und Information wird implizit deutlich. Die Schüler wandeln Daten, die sie von Sensoren erhalten, in sinnvolle Informationen um. Sie interpretieren analoge und digitale Eingabewerte. Bei der Verarbeitung von Daten verwenden die Schüler arithmetische und logische Operationen. Sie sind außerdem in der Lage, Informationen zu Daten zu kodieren, welche an die Aktoren ausgegeben werden können, so dass das gewünschte Verhalten eintritt.
- 2) **Algorithmen:** Beim Entwurf ihrer interaktiven Objekte und Installationen formulieren die Schüler selbstständig Algorithmen, um zuvor gestellte Ziele zu verwirklichen. Die entworfenen Algorithmen werden unter Nutzung der durch Sensoren gewonnenen Daten und der durch Aktoren realisierten Aktionen von den Schülern getestet und gegebenenfalls modifiziert.
- 3) **Informatiksysteme:** Ein Verständnis von (eingebetteten) interaktiven Informatiksystemen wird durch die Arbeit mit dem Physical Computing-Baukasten erworben. Die Schüler erstellen mit diesem Werkzeug selbstständig neue Informatiksysteme.

Prozessbereiche:

- 1) **Modellieren und Implementieren:** Die Schüler implementieren in ihrer Ganzheit zu betrachtende Informatiksysteme bestehend aus Hard- und Software und beurteilen ihre Entwürfe kritisch. Sie nehmen bewusst Veränderungen an ihren Implementationen und an der Hardware vor und beobachten die dabei entstehenden Auswirkungen.
- 2) **Begründen und Bewerten:** Bei der Konzeption ihrer Projekte nutzen die Schüler ihr informatisches Wissen, um komplexe Problemstellungen zu entwerfen. Sie stellen Vermutungen über geeignete Lösungswege an und treffen begründete Entscheidungen zur Modellierung informatischer Sachverhalte.

5 Physical Computing mit „My Interactive Garden“

Der Einsatz von Physical Computing im Informatikunterricht wirft Fragen hinsichtlich einzusetzender Hardware, Programmierumgebung und des Unterrichtskontexts auf. My Interactive Garden (MyIG) [PR12] stellt eine prototypische Gestaltungs-, Programmier- und Lernumgebung dar, mit welcher Potential und Grenzen des Physical Computing im Informatikunterricht untersucht werden sollen. Im Folgenden werden wesentliche Designentscheidungen sowie erste Unterrichtserfahrungen diskutiert.

MyIG versucht eine Brücke zwischen den oben charakterisierten Mikrocontrollerboards und vorgefertigten Sensoren und Aktoren zu schlagen. Hierfür ist es notwendig, die Hardware entsprechend zu vereinfachen, welche unter Berücksichtigung der folgenden Prinzipien realisiert wurde:

- 1) **Einfache Handhabung:** Die Schüler erhalten vorgefertigte Sensoren und Aktoren mit einfachen Steckverbindungen, die kein Hantieren mit losen Litzen oder gar Löten erfordern, sondern sofortiges Experimentieren ermöglichen.
- 2) **Flexibilität und Erweiterbarkeit:** Die Komponenten sollten eine flexible Gestaltung interaktiver Objekte ermöglichen. Bei Bedarf soll eine unkomplizierte Erweiterung um zusätzliche Komponenten möglich sein.
- 3) **Black-/Whiteboxen:** Komponenten, die für das Verständnis des Sensors oder Aktors irrelevant sind, müssen nicht sichtbar sein, sollen aber bei Bedarf sichtbar werden können.
- 4) **Betonung informatischer Grundsätze:** Grundlegende informatische Prinzipien sollen deutlich werden, z. B. das EVA-Prinzip durch optische Trennung der Anschlüsse für Ein- und Ausgabekomponenten (Sensoren/Aktoren).

MyIG stellt auf der Basis von Arduino einen Baukasten zur Verfügung, der neben vorgefertigten Sensoren und Aktoren auch ein Shield enthält, welches ein einfaches Anstecken der Komponenten ermöglicht (Abb. 2). Zur Programmierung können damit verschiedene Programmiersprachen genutzt werden, wie z. B. eine Variante von Scratch (S4A) (Abb. 3) oder Arduino-C. Hinzu kommen beliebige zusätzliche Materialien, z. B. ein Bastelset.

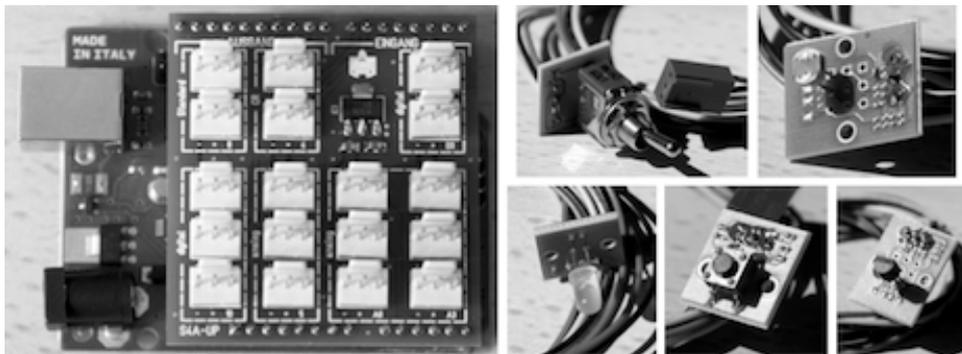


Abbildung 2: Arduino mit aufgestecktem MyIG-Shield und vorgefertigte Sensoren und Aktoren

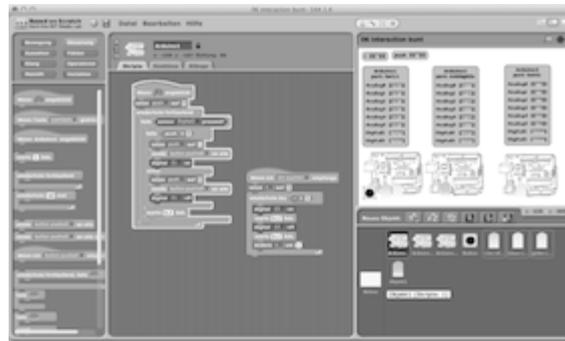


Abbildung 3: Programmierumgebung S4A

Ziel des Lernens mit MyIG ist die kollaborative Erstellung einer Ausstellung interaktiver Objekte, wie sie in einem futuristischen interaktiven Garten zu finden sein können – ein Rahmen, der vielfältige Projekte ermöglicht und die Kreativität der Schüler anregen soll. Folgende Strategien werden in Anlehnung an [Ru08] empfohlen:

- Konzentration auf Themen statt Aufgaben
- Kombination von Kunst und informatischem Modellieren
- Ermutigung zum Erzählen von Geschichten
- Organisation von Ausstellungen statt Wettbewerben

Erprobung

In einem ersten Schulversuch wurde der Einsatz des MyIG-Baukastens in einer neunten Klasse erprobt. Möglichkeiten und Grenzen von Physical Computing wurden hinsichtlich der folgenden Aspekte untersucht: (1) Wahrnehmung eingebetteter Informatiksysteme, (2) Akzeptanz „informatischen Töpfers“, (3) Ausgewogenheit informatischer und ergänzender Basteltätigkeiten, (4) Mehrwert des Physical Computing.



Abbildung 4: Beispielprojekte „Interaktive Blume“ und „Lagerfeuer“

Zur Motivation und Darstellung der Möglichkeiten von Physical Computing, wurden zu Beginn des Schulversuchs zwei Beispielprojekte vorgestellt (Abb. 4), gefolgt von einer Experimentierphase, in der die Schüler eigenständig alle Komponenten des Baukastens ausprobieren konnten. Im Anschluss daran erstellten die Schüler einen Projektplan für ihre ersten interaktiven Objekte zum Thema „My Interactive Garden“. Während der Arbeitsphase arbeiten sie nach ihren selbst erstellten Zeitplänen. In regelmäßigen Evaluationsphasen wurden Fortschritte reflektiert, mögliche Probleme und Lösungs-

ansätze diskutiert und fachliche Erkenntnisse genutzt, um den Zeitplan an die sich neu ergebenden Umstände anzupassen.

(1) Zu Beginn der Erprobung nannten die Schüler auf die Frage, wo Informatik überall eine Rolle spielt, überwiegend Beispiele, die sich unter dem Begriffen „Standardsoftware“, „Computerspiele“ und „html“ subsumieren lassen. Zwei Schüler nannten außerdem LEGO-Mindstorms-Roboter und Smartphones. Eingebettete Systeme des Alltags hingegen sind ihnen als typische Produkte der Informatik nicht eingefallen. Im Anschluss an die Experimentierphase war schon ein erster Wandel dieser Informatikwahrnehmung zu beobachten. Während sich die Schüler darüber austauschten, was sie über die Funktionsweise der Sensoren und Aktoren herausgefunden haben, waren häufig Vergleiche wie: „Das ist wie beim Auto, da piept es auch unterschiedlich schnell, je nachdem wie weit ein Hindernis entfernt ist.“ zu vernehmen.

(2) Mittels eines Fragebogens wurden die teilnehmenden Schüler vorab zu ihrer Wahrnehmung des Informatikunterrichts befragt. Dabei stellte sich heraus, dass sie einstimmig nicht das Gefühl hatten, im Informatikunterricht gemeinsam an etwas Größerem arbeiten zu können. Gleichzeitig antworteten einige Schüler positiv auf die Frage, ob sie Produkte des Informatikunterrichts bereits einmal anderen vorgeführt haben. Dies erlaubt die Schlussfolgerung, dass durchaus ein Interesse daran besteht, im Informatikunterricht vorzeigbare Produkte zu fertigen. Im Projektverlauf erschuf nun jeder Schüler sein eigenes interaktives Objekt, welches dann am Ende in die gemeinsame Ausstellung des interaktiven Gartens integriert wurde.

(3) Es besteht die Gefahr, dass beim Physical Computing das Basteln in den Vordergrund und die informatischen Inhalte in den Hintergrund rücken. Unsere Vermutung ist jedoch, dass umfangreiches Basteln auch umfangreiche Programmierung implizieren kann, da das Gesamtwerk an Komplexität gewinnt. Interessanterweise ist in der Lerngruppe den meisten Schülern die Programmierung wichtiger als das Basteln. Die Schüler versuchen, in ihren Projekten so viele Sensoren und Aktoren unterzubringen wie möglich und umfangreiche Programme zu erstellen. Die äußere Hülle des interaktiven Objekts erscheint derzeit noch eher nebensächlich zu sein, die Schüler sind eher von der Technik, also den Sensoren und Aktoren, fasziniert als vom Bastelmaterial. Gleichzeitig konnten wir aber auch feststellen, dass sie sich darauf freuen, ihren Erfindungen die passende Form zu geben. Sie haben zahlreiche kreative Ideen - nur ist dieser Aspekt nicht prioritär in der Reihenfolge der auszuführenden Arbeitsschritte in der Projektplanung.

(4) Im Rahmen der Physical-Computing-Projekte konnten relevante Fragestellungen an konkreten Problemen untersucht werden, die sich sonst im Informatikunterricht oft nur abstrakt bearbeiten lassen: Was ist der Unterschied zwischen analog und digital? Welche Komponenten gehören an die Eingänge, welche an die Ausgänge? Größtenteils konnten sich die Schüler ihre Fragen selbst oder gegenseitig beantworten. Interessant war zu beobachten, dass sie sich nicht lange damit aufhielten, die einzelnen Sensoren und Aktoren nacheinander zu testen, sondern sie schnell in umfangreichere Gebilde eingebunden haben. So entstand beispielsweise bei einem Schüler schon in der Experimentierphase eine komplexe Ampelschaltung mit durch Helligkeitssensor

gesteuerte Tag- und Nachtmodi, Bedarfsschaltung und Piepton für blinde Fußgänger. Ein anderer Schüler entwarf ein komplexes Gebilde aus Taster, Kippschalter, blinkenden Lampen, Summer und Servo. Alle Schüler sind im Laufe der Experimentierphase mit dem Problem der Mehrfachausführung konfrontiert gewesen, haben die „forever“-Schleife und bedingte Schleifen kennengelernt und sich mit Alternativen befasst. Arithmetische Operationen, Botschaften und Variablen wurden ebenfalls bereits verwendet – das alles geschah automatisch, intuitiv und interessengesteuert und ohne aktives Eingreifen durch die Lehrperson. Auffällig war auch, dass die Schüler offensichtlich den experimentellen Zugang bevorzugten. Sie haben erst dann nach Hilfe gefragt, wenn sie allein tatsächlich nicht mehr weiter kamen. Dies war jedoch nur selten der Fall.

6 Diskussion und Ausblick

Informatikunterricht steht seit jeher vor der Herausforderung, zeitstabile Ideen und Konzepte zu vermitteln, diese aber an aktuellen Themen und unter Zuhilfenahme moderner Werkzeuge zu motivieren. Mit Physical Computing eröffnet sich für den Informatikunterricht ein Anwendungsgebiet, welches so wie Mindstorms die Ideen von Logo erfolgreich in die reale Welt übertragen hat, das Potential der kreativen Gestaltung von Scratch in Produkte der realen Welt transferiert. Betrachtet man den Motivationswert eines selbstkreierten Produkts, wie z. B. durch Töpfern, wird schnell deutlich, dass dieses sich auch für den Informatikunterricht nutzen lässt. Die Verbindung von Informatik mit Kunst und Design hat das Potential, auch weniger computeraffine Schüler anzusprechen.

Unsere Erfahrungen zeigen, dass Schüler Spaß haben am Physical Computing. Dies äußerten sie verbal, zeigten es aber auch beispielsweise dadurch, dass sie am Ende des Unterrichts nicht nach Hause gehen wollten und stattdessen noch weiter mit den Bauteilen experimentierten. Die Schüler wünschten sich außerdem Baukästen für zu Hause, da sie sonst nur einmal pro Woche Gelegenheit haben, an ihren interaktiven Objekten zu arbeiten.

Während es sich bei MyIG um einen lokal hergestellten Prototypen handelt, befinden sich zur Zeit verschiedene vergleichbare Systeme in der Entwicklung, die in absehbarer Zeit für den Einsatz im Unterricht erhältlich sein sollen (z. B. [TK13], [Go10]). Die ersten Erfahrungen mit MyIG zeigen, dass Physical Computing im Informatikunterricht einen spannenden und vielversprechenden Ansatz darstellt. Wir gehen davon aus, dass in wenigen Jahren die Erstellung interaktiver Objekte einen festen Platz in der informatischen Bildung eingenommen haben wird, ähnlich wie Mindstorms-Robotik heute. Dieser Beitrag kann für den Einsatz der Systeme eine erste Orientierung geben.

Literaturverzeichnis

- [Ba11] Banzi, M.: Getting Started with Arduino. O'Reilly Media / Make, Sebastopol, CA, 2011.
- [Bau11] Baumann, R.: Eingebettete Systeme verstehen. In: LOG IN, H. 171, LOG IN Verlag GmbH, 2011; S. 33-45.

- [GI08] Gesellschaft für Informatik (GI) e.V.: *Grundsätze und Standards für die Informatik in der Schule*. In: LOG IN, H. 150/151, LOG IN Verlag GmbH, 2008; Beilage.
- [Go10] Gomba, D.: Visual Programming Arduino: modkit and the others. 2010. <http://arduino.cc/blog/2010/10/05/visual-programming-arduino-modkit-and-the-others/> (07.03.2013)
- [Gu10] Guzdial, M.: Dancing and singing humans, even more than robots. 2010. <http://computinged.wordpress.com/2010/12/31/> (07.03.2013).
- [Kn11] Knobelsdorf, M.: Biographische Lern- und Bildungsprozesse im Handlungskontext der Computernutzung. Dissertation, Berlin, 2011.
- [Ma12] Petre, M.: Computing Is Not a Spectator Sport: Rethinking How We Introduce Our Discipline to Students. In: GI (Hrsg.): Praxisband zur GI-Fachtagung „HDI 2012 – Informatik für eine nachhaltige Zukunft“. Universitätsverlag Potsdam, 2012. S. 155-159.
- [MKL10] Marcu, G.; Kaufman, S. J.; et al: Design and evaluation of a computer science and engineering course for middle school girls. Proc. SIGCSE 2010, Milwaukee, ACM, 2010.
- [PA13] Pelz, L.; Arnhold, W.: Die Waschmaschine - Embedded Computing im Alltag. 2013. Workshop auf der 12. Tagung der GI-Fachgruppe „Informatik-Bildung in Berlin/BRB“.
- [PH91] Papert, S., Harel, I. Situating Constructionism. In: Papert, S.; Harel, I. (Hrsg.), Constructionism. Ablex Publishing Corporation. Norwood, 1991.
- [PIC08] The Playful Invention Company (PICO): PicoCricket – Invention kit that integrates art and technology. 2008. <http://www.picocricket.com/whatisit.html> (06.03.2013)
- [PIC10] The Playful Invention Company (PICO): Picoboard – Sensor Board that works with MIT's Scratch. 2010. <http://www.picocricket.com/picoboard.html> (06.03.2013)
- [PR12] Przybylla, M.; Romeike, R.: My Interactive Garden – A Constructionist Approach to Creative Learning with Interactive Installations in Computing Education. In: Kynigos, C.; Clayson, J. E.; Yiannoutsou, N.: Constructionism: Theory, Practice and Impact. Proceedings of Constructionism 2012. Athen. S. 395-404.
- [Re07] Resnick, M.: Sowing the Seeds for a More Creative Society. Proc. Learning & Leading with Technology, International Society for Technology in Education (ISTE), 2007.
- [RR11] Romeike, R.; Reichert, D.: PicoCrickets als Zugang zur Informatik in der Grundschule. In: Thomas, M. (Hrsg.): Informatik in Bildung und Beruf (14. GI Fachtagung Informatik und Schule - INFOS 2011). Köllen, Bonn 2011, S. 177-186.
- [RS10] Richards, M.; Smith, N.: Teaching UbiComp with Sense. In: Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordCHI '10), ACM, New York, 2010; S. 765-768.
- [Ru08] Rusk, N., Resnick, M., Berg, R., & Pezalla-Granlund, M.: New Pathways into Robotics: Strategies for Broadening Participation, In: Journal of Science Education and Technology, 17 (1), 2008; S. 59-69.
- [St09] Strecker, K.: Informatik für Alle – Wie viel Programmierung braucht der Mensch? Dissertation, Göttingen, 2009.
- [St11] Strecker, K.: Wie viel Programmierkompetenz braucht der Mensch? In: LOG IN, H. 169/170, LOG IN Verlag GmbH, 2011; S. 40-47.
- [TK13] TinkerKit: TinkerKit! 2013. <http://www.tinkerkit.com/> (07.03.2013)
- [Va12] Vahrenhold, Jan: On the Importance of Being Earnest: Challenges in Computer Science Education. Proceedings of WiPSCe 2012, ACM, New York, 2012.
- [Wa05] Wagner, O.: LEGO Roboter im Informatikunterricht - Eine Untersuchung zum Einsatz des LEGO-Mindstorms-Systems zur Steigerung des Kooperationsvermögens im Informatikunterricht eines Grundkurses (12. Jahrgang, 2. Lernjahr) der Otto-Nagel-Oberschule (Gymnasium). Schriftliche Prüfungsarbeit im Rahmen der zweiten Staatsprüfung für das Amt des Studienrates, Berlin, 2005.
- [We08] Weber, M.: Vermittlung von informatischen Grundkonzepten der Realschulbildung anhand einer robotergesteuerten Lagerverwaltung. Universität Erlangen-Nürnberg, Erlangen, 2008.

Belebt die Einführung des Schulfachs Informatik die Nachfrage nach einem Informatikstudium?

Eine empirische Langzeitstudie unter besonderer Berücksichtigung des Vergleichs der Absolventen von alter (G9) und neuer gymnasialer Oberstufe (G8) in Bayern

Christian Götz

Studien-Service-Center Informatik
Universität Erlangen-Nürnberg
Martensstr. 3
91058 Erlangen
christian.goetz@fau.de

Peter Hubwieser

TUM School of Education
Technische Universität München
Marsstr. 20-22
80335 München
peter.hubwieser@tum.de

Abstract: Im bundesweiten Vergleich nahm die Zahl der Informatikstudierenden in den letzten zwei Jahren in Bayern überdurchschnittlich zu. Diese Entwicklung könnte mit dem doppelten Abiturjahrgang 2011 und der Einführung des Schulfachs Informatik im 8-stufigen Gymnasium in Bayern zusammenhängen. Diese Arbeit zeigt Ergebnisse einer Studie, welche bei Studienanfängern durchgeführt wurde, um gezielt Unterschiede zwischen G9- und G8-Absolventen in Bezug auf den Einfluss des Schulfachs Informatik auf die Studienwahl herauszufinden.

1 Einleitung

Das Statistische Bundesamt meldet einen Anstieg der Anzahl der Studierenden in Deutschland auf 2.377.034 im Wintersemester (WS) 2011/12, was einer Zunahme von 7,2 % im Vergleich zum Vorjahr und über 24 % im Vergleich zum Stand vor drei Jahren entspricht. Auch im Bundesland Bayern erreichte die Zahl der Studierenden im WS 2011/12 mit erstmals 320.327 Studierenden (11,4 % mehr als im Vorjahr) einen neuen Höchststand. Leszczensky et. al. sehen die zentrale Ursache für den allgemeinen Anstieg der Studierendenzahlen in der wachsenden Beteiligung der altersgleichen Bevölkerung an zur Hochschulreife führender Schulbildung (einschl. beruflicher Bildung). Des Weiteren werden Erleichterungen im Bereich der Hochschulzugänge für Techniker, Meister, fachgebundene Hochschulberechtigte, die Aussetzung der Wehrpflicht sowie die doppelten Abiturientenjahrgänge als Ursachen aufgeführt [Le12]. Auch die Zahl der bundesweit immatrikulierten Informatikstudierenden stieg sprunghaft von 69.559 (WS 2010/11) auf 76.098 (WS 2011/12) an, was einer Zunahme von 9,4 % entspricht. In Bayern stieg die Zahl der Informatikstudierenden im gleichen Zeitraum um fast 15 % von 7.412 (WS 2010/11) auf 8.505 (WS 2011/12) (alle Angaben Statistisches Bundesamt). Diese im bundesweiten Vergleich überdurchschnittliche Zunahme legt die Vermutung nahe, dass diese Entwicklung auch mit der Einführung des Schulfachs Informatik im 8-stufigen Gymnasium in Bayern in Verbindung stehen könnte.

In den letzten Jahren wurden auf europäischer- und internationaler Ebene zahlreiche Diskussionen über die Einführung eines Schulfachs Informatik in der Sekundarstufe geführt (ein guter Überblick findet sich z. B. bei [Hu12, S. 5]). An bayerischen Gymnasien wurde mit der Umstellung auf das 8-stufige Gymnasium (kurz G8) ab dem Schuljahr 2004/2005 das Schulfach Informatik verpflichtend ab der 6. Jahrgangsstufe eingeführt (vgl. [Hu07] [Hu05]). Im Jahr 2011 konnte der doppelte Abiturjahrgang in Bayern (letzter 9-stufiger, kurz G9, und erster 8-stufiger Abiturjahrgang 2011) an den Universitäten und Hochschulen zu studieren beginnen. Um die Studienbedingungen für den Doppelabiturjahrgang zu verbessern, wurde der Abiturtermin für den letzten G9-Jahrgang vorverlegt und zusätzlich in Bayern ein Studienbeginn zum Sommersemester (SS) 2011 ermöglicht.

2 Motivation und Zielsetzung der Studie

Diese Arbeit ist Teil eines umfangreicheren Forschungsprojekts an der Technischen Fakultät der Universität Erlangen-Nürnberg mit dem Ziel, wichtige Ansatzpunkte zu finden, um mehr Schülerinnen und Schüler für ein Informatikstudium zu gewinnen. Die Motivation zu dieser Arbeit begründet sich nicht zuletzt auf der Tatsache, dass trotz steigender Studierendenzahlen die Absolventen im MINT-Bereich und speziell in der Informatik nicht ausreichen werden, den IT-Fachkräftemangel langfristig zu decken (vgl. z. B. [AKP12], [PR11], [KP09]). Bedingt durch den demografischen Wandel prognostiziert eine Studie bereits für 2015, dass ca. 14 Prozent der insgesamt angebotenen Stellen für Ingenieure und Naturwissenschaftler nicht besetzt werden können (vgl. [PR11, S. 35]). Bisherige Studien belegen, dass Schülerinnen und Schüler meist falsche bzw. klischeehaft geprägte Vorstellungen von Informatikerinnen und Informatikern und deren möglichen beruflichen Tätigkeitsfeldern haben und sie deshalb andere Studiengänge und Berufe vorziehen (vgl. z. B. [BD08], [MW06], [Ca06]). Eine Studie der TU-München kommt zu dem Ergebnis, dass gerade mathematisch interessierte und informatikgeeignete Schülerinnen und Schüler kein Informatikstudium in Betracht ziehen. Die gleiche Studie sieht aber positive Effekte und Einflussmöglichkeiten durch einen Informatik-Schulunterricht [En08]. Nach [Kn11] gründet sich das Interesse und die Intention einer möglichen Studienwahl Informatik anhand unterschiedlicher biographischer Lern- und Bildungsprozesse im frühzeitigen Umgang mit IT-Systemen. Aus unterschiedlichen (meist intrinsisch geleiteten) Motiven und Vorerfahrungen entstehen so auch unterschiedliche Erwartungshaltungen der Schülerinnen und Schüler gegenüber dem Informatikunterricht. Magenheimer meint, dass dem Informatikunterricht deshalb „[...] eine homogenisierende und aufklärerische Funktion im Hinblick auf informatische Problemstellungen und computerbezogene berufliche Tätigkeitsfelder zukommen sollte“ (vgl. [MS05, S.121]). Zusammenfassend bescheinigen die Studien dem Unterrichtsfach Informatik einen hohen, motivationalen Stellenwert und lassen einen Zusammenhang für eine spätere Präferenz und Wahl eines Informatikstudiums vermuten.

Auf Basis der bisherigen Erkenntnisse stellt sich nun die Frage, in wie fern sich der Einfluss eines verpflichtenden Schulfachs Informatik nachweisen lässt. Hierzu wurde an der Universität Erlangen-Nürnberg im Zeitraum vom SS 2011 (letzter G9 Jahrgang) bis WS

2012/13 eine Studie unter allen Erstsemestern der Informatikstudiengänge durchgeführt, um gezielt Unterschiede zwischen Absolventen des G9 und G8 festzustellen.

3 Studie zu Unterschieden zwischen G9- und G8-Absolventen

3.1 Design der Studie

In der seit SS 2011 durchgeführten Studie wurden alle Erstsemester der Bachelorstudiengänge Informatik (Inf), Informations- und Kommunikationstechnik (IuK), Computational Engineering (CE) und Elektrotechnik-, Elektronik-, Informationstechnik (EEI - als Vergleichsgruppe) an der Universität Erlangen-Nürnberg per personalisierter E-Mail-Einladung in der ersten Studienwoche gebeten, an der Online-Befragung teilzunehmen. Der Teilnahmezeitraum war auf 3 Wochen begrenzt, Nichtteilnehmer wurden zweimalig zur Teilnahme per E-Mail aufgefordert. Für die Auswertung wurden nur vollständig ausgefüllte Fragebögen von Studierenden mit bayerischer Hochschulzugangsberechtigung aus dem G8, bzw. G9 gezählt. Andere Hochschulzugangsberechtigungen gingen nicht in die Auswertung mit ein. Die Gesamtrücklaufquote der Stichprobe betrug gemittelt über beide Kohorten (G9- und G8-Absolventen) ca. 39 % (vgl. Tabelle 1).

Semester	Grundgesamtheit Erstsemester (Inf, IuK und CE)	davon		Rücklauf (Quote)	
		Grund- gesamtheit G9	Grund- gesamtheit G8	G9	G8
SS 2011	167	138	0	62 (37,1 %)	-
WS 2011/12	237	109	100	37 (33,9 %)	43 (43,0 %)
SS 2012	78	45	14	9 (20,0 %)	4 (28,5 %)
WS 2012/13	236	42	158	15 (35,7 %)	67 (42,4 %)
Gesamt Inf	718	334	272	123 (36,8 %)	114 (41,9 %)
Vergleichs- gruppe EEI	EEI gesamt 502	209	211	90 (43,1 %)	69 (32,7 %)

Tabelle 1: Grundgesamtheit und Rücklauf der Befragung

Der Online-Fragebogen zum Informatikunterricht bestand aus sechs Fragen. Als erstes wurde danach gefragt, ob überhaupt Informatikunterricht stattfand. Als zweites wurde gefragt, wie interessant der Informatikunterricht subjektiv empfunden wurde. Hier gab es vier Antwortalternativen: „gar nicht interessant“, „eher weniger interessant“, „eher interessant“ und „sehr interessant“. Anschließend wurde nach dem Neuigkeitsgehalt des Informatikunterrichts gefragt. Hierzu konnte in 20 %-Schritten gewählt werden, wie viel Prozent des Informatikunterrichts für die Schülerinnen und Schüler neue Inhalte, bzw. Wissen bereit hielt. Die unterste Ausprägung (0 %) entsprach dabei der Aussage, dass im Informatikunterricht für den entsprechenden Schüler kein neues Wissen vermittelt wurde, die oberste Ausprägung (100 %) entsprach der Aussage, dass jeglicher Inhalt als neues Wissen eingestuft wurde. Die vierte Frage bezog sich auf das Vorwissen der Schülerinnen und Schüler bzw. woher die Schülerinnen und Schüler den Großteil ihrer IT-Kenntnisse erworben hatten. Hierzu konnten (Mehrfachauswahl war möglich) die Optionen „Mit Freunden, Bekannten, Experten zusammen erarbeitet“, „Zu Hause von Eltern-

teil oder Geschwister gelernt“, „Zu Hause aus Internetportalen selbst angeeignet“, „Zu Hause aus Büchern/Zeitschriften etc. selbst angeeignet“ und „In der Schule (z. B. im Informatikunterricht) gelernt“ gewählt werden. Als Fünftes wurde danach gefragt, ob mehr schulische oder außerschulische Faktoren ausschlaggebend für die Studienwahl waren. Hierzu gab es drei Antwortalternativen: „mehr außerschulische Faktoren“, „beides gleich“ oder „mehr schulische Faktoren“. Als Letztes wurde nach der Beeinflussung der Studienwahl durch den Informatikunterricht gefragt. Hierzu gab es die Auswahlmöglichkeiten „hat mich positiv beeinflusst“, „hat mich gar nicht beeinflusst“ und „hat mich negativ beeinflusst“.

3.2 Ergebnisse

Frage 1: Hatten Sie in der Schule Informatikunterricht?

Von den $N = 123$ Antworten der G9-Absolventen, welche sich für ein Informatikstudium entschieden haben, gaben nur $n = 30$ (24,4 %) an, dass sie Informatikunterricht hatten. In der Vergleichsgruppe der Elektrotechnik-Erstsemester ($N = 90$) gaben $n = 28$ (31,1 %) an, dass sie im G9 Informatikunterricht hatten. Bei den Absolventen des G8 mit Studienwahl Informatik ($N = 114$) gaben $n = 98$ (86,0 %) an, dass sie Informatikunterricht hatten. In der Vergleichsgruppe der EEI Studienanfänger aus dem G8 ($N = 69$) hatten $n = 62$ (89,9 %) Informatikunterricht. Im bayerischen G8 gibt es verpflichtenden Informatikunterricht für alle Schülerinnen und Schüler in der 6. und 7. Jahrgangsstufe im Rahmen des Fächerverbundes Natur und Technik. Darüber hinaus findet weiterführender Informatikunterricht nur im naturwissenschaftlich-technologischen Zweig in den Jahrgangsstufen 9 bis 12 statt. Daraus erklärt sich die nicht ganz 100-prozentige Abdeckung in der Umfrage (vgl. Abb. 1).

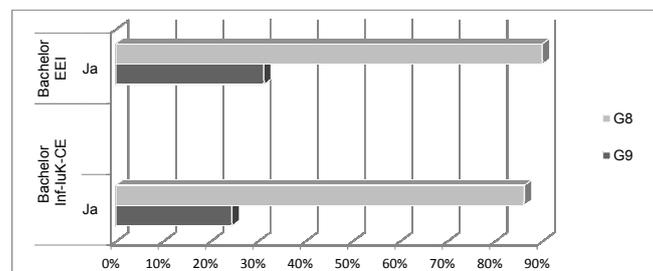


Abbildung 1: Erstsemester mit Informatikunterricht - Vergleich G9- und G8-Absolventen

Frage 2: Wie interessant fanden Sie den Informatikunterricht?

Die Auswertung dieser Frage zeigte große Unterschiede, sowohl zwischen den Erstsemestern der einzelnen Studiengänge, als auch bei den Kohorten der G9- und G8-Absolventen (vgl. Abb. 2). Generell wurde der Interessantheitsgrad des Informatik-

unterrichts von den Erstsemestern der Informatikstudiengänge besser bewertet als von den Erstsemestern der EEI-Vergleichsgruppe. In der Kohorte der G8-Absolventen ($N = 98$) der Informatikerstsemester gaben $n = 84$ (85,7 %) an, dass sie den Informatikunterricht als sehr, bzw. eher interessant erlebt hatten. Bei den G9-Absolventen ($N = 30$) waren dies nur $n = 17$ (56,6 %). In der Vergleichsgruppe der EEI Erstsemester gaben bei den G8-Absolventen ($N = 62$) nur $n = 30$ (48,4 %) die Einstufung als sehr/eher interessant, bei den G9-Absolventen ($N = 28$) nur $n = 15$ (53,6 %) an. Die Grafik zeigt auch, dass es in der Kohorte der G9-Absolventen (sowohl bei den Informatik- als auch bei den EEI-Studienbeginnern) zwei Ausprägungen bei der Einschätzung des Interessanztheitsgrades des Informatikunterrichts im G9 gab: Ein Peak (in beiden Studiengruppen), welche den Unterricht als „eher uninteressant“ empfanden und ein anderer, welche bei der Einstufung „sehr interessant“ lag. Diese Unterschiede waren in der Kohorte der G8-Absolventen nicht vorhanden.

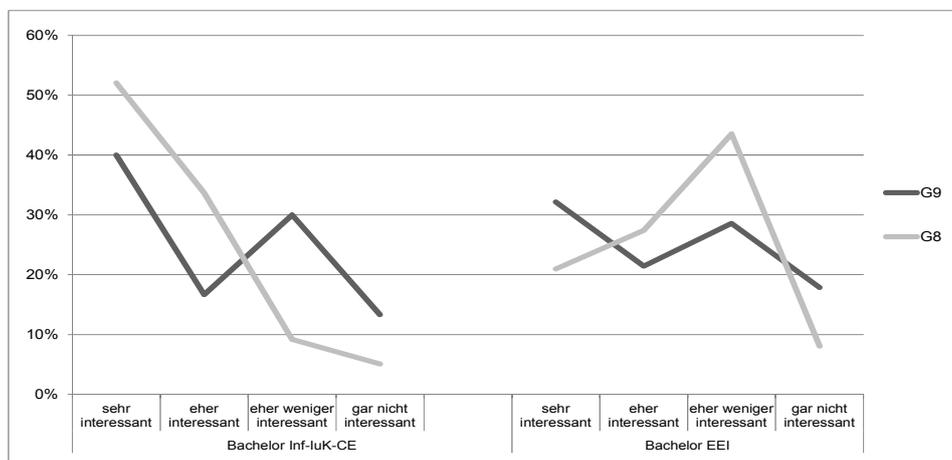


Abbildung 2: Subjektive Einschätzung des Interessanztheitsgrades des Informatikunterrichts

Frage 3: Wieviel Prozent vom Inhalt des Informatikunterrichts waren für Sie neu?

Bei dieser Frage zeigten sich bei den Erstsemestern der Informatikstudiengänge große Unterschiede zwischen G9- und G8-Absolventen (vgl. Abb. 3). In der Kohorte der G9-Absolventen ($N = 30$) gab es eine Häufung bei den Ausprägungen 0 % bis 40 % (entsprechend kaum neue Wissensinhalte, $n_k = 16$; 53,3 %) und den Ausprägungen 80 % und 100 % (entsprechend fast alles im Unterricht wurde als neues Wissen eingestuft) $n_k = 10$ (33,3 %). Bei den G8-Absolventen ($N = 98$) entfielen die meisten Antworten auf die Ausprägungen 60 % und 80 % ($n_k = 57$; 68,2 %). In der Vergleichsgruppe der EEI-Studierenden gab es dagegen kaum Unterschiede zwischen G9- und G8-Absolventen. In beiden Kohorten wurde der Neuigkeitsgehalt des Informatikunterrichts meist bei 60 % und 80 % gesehen (G9: $n_k = 13$; 46,4 %, G8: $n_k = 36$; 58,1 %)

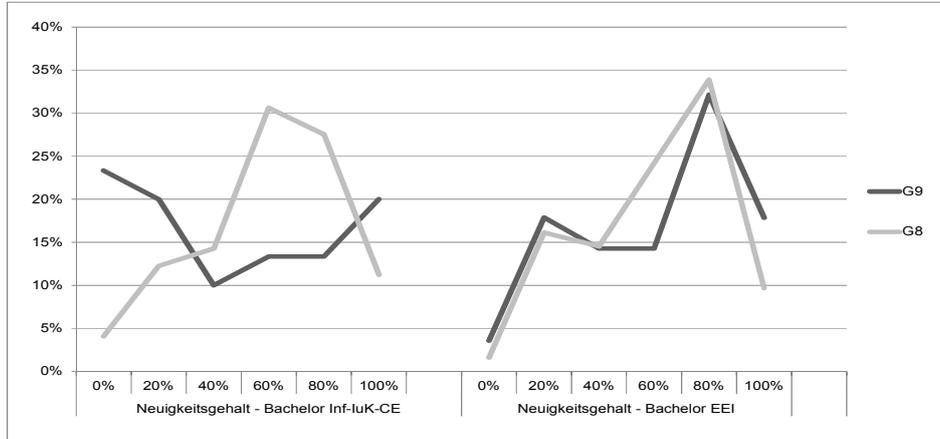


Abbildung 3: Einschätzung des Neuigkeitsgehaltes des Informatikunterrichts
 n_k = kumulierte Werte entsprechend der Ausprägungsangaben

Frage 4: Woher haben Sie den Großteil ihrer IT-Kenntnisse erworben?

Die Auswertung dieser Frage (vgl. Abb. 4) ergab in der Vergleichsgruppe der EEI-Erstsemester bis auf die letzte Antwortvariante kaum Unterschiede zwischen G9- und G8-Absolventen. Hier zeigte sich bei den G8-Absolventen im Bereich der durch die Schule erworbenen Kenntnisse eine Verdoppelung der Angaben im Vergleich zu den G9-Absolventen. In der Gruppe der Erstsemester der Informatikstudiengänge zeigte sich bei Kohorte der G8-Absolventen bei der durch die Schule erworbenen Kenntnisse eine Zunahme um mehr als das 5-fache im Vergleich zu den G9-Absolventen. Trotz dieses enormen Zuwachses im schulischen Bereich gab es in dieser Gruppe bei den Items z. B. in Bezug zum Eigenstudium kaum Unterschiede zwischen den Kohorten der G9- und G8-Absolventen.

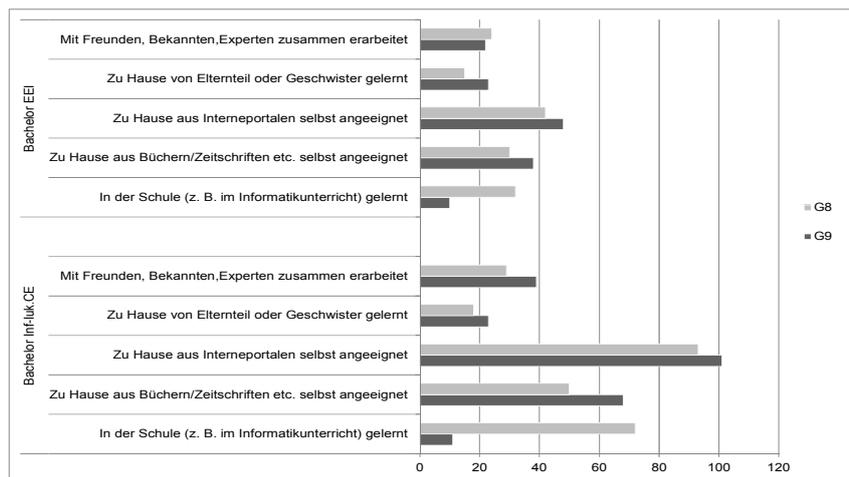


Abbildung 4: Wo wurde der Großteil des IT-Wissens erworben? Nennungen absolut.

Frage 5: Waren mehr schulische oder außerschulische Faktoren ausschlaggebend für die Studienwahl?

Während sich innerhalb der Vergleichsgruppe der Elektrotechnik-Erstsemester bei dieser Frage kaum Unterschiede zwischen G9- und G8-Absolventen zeigten, ergab die Auswertung bei den Erstsemestern der Informatikstudierenden eine deutliche Tendenz (vgl. Abb. 5). In der Kohorte der G9-Absolventen ($N = 123$) gaben $n = 109$ (88,6 %) an, dass mehr außerschulische Faktoren ausschlaggebend für ihre Studienwahl waren. Nur $n = 2$ (1,62 %) gaben schulische Faktoren an. Bei den G8-Absolventen mit Studienwahl Informatik ($N = 114$) gaben $n = 64$ (56,1 %) außerschulische Faktoren an, $n = 15$ (13,1 %) sahen mehr schulische Faktoren als Einflussfaktor und $n = 35$ (30,7 %) meinten beide Faktoren hätten gleich viel Einfluss auf ihre Studienwahl gehabt.

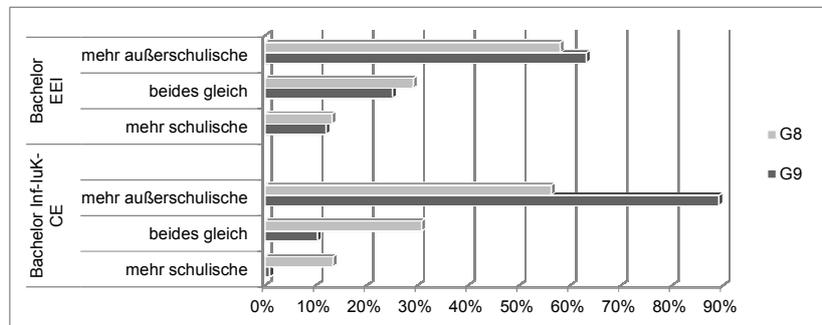


Abbildung 5: Schulischer oder außerschulischer Einfluss auf die Studienwahl.

Frage 6: Beeinflussung der Studienwahl durch den Informatikunterricht

In der Vergleichsgruppe der EEI-Studierenden gab es bei dieser Frage wiederum kaum Unterschiede zwischen G9- und G8-Absolventen. Die meisten gaben an, dass sie durch den Informatikunterricht gar nicht beeinflusst wurden (vgl. Abb. 6). In der Gruppe der Informatik-Erstsemester gaben von den G9-Absolventen, welche Informatikunterricht hatten ($N = 30$) je $n = 15$ (50,0 %) an, dass sie der Informatikunterricht positiv, bzw. gar nicht beeinflusst hat. Bei den G8-Absolventen ($N = 98$) gaben $n = 82$ (83,7 %) an, dass sie der Informatikunterricht positiv beeinflusst hat.

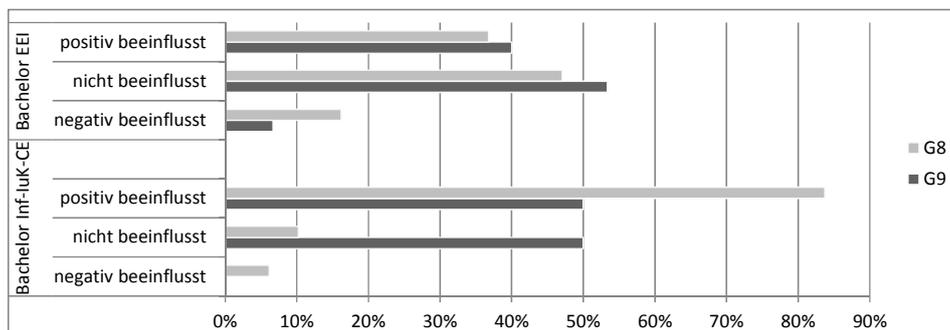


Abbildung 6: Einfluss des Informatikunterrichts auf die Studienwahl.

3.3 Diskussion

Aus den Ergebnissen dieser Studie kann geschlossen werden, dass die Einführung des Schulfachs Informatik erhebliche Auswirkungen auf die Studienwahl Informatik hat. Da es im G9 keinen curricular verankerten Informatikunterricht gab, kann aus dem Ergebnis der Frage 1 gedeutet werden, dass im G9 nur vereinzelt und lokal unterschiedlich Informatik unterrichtet wurde. Aus den Fragen 2 und 3 kann z. B. abgeleitet werden, dass es bei den G9-Absolventen scheinbar zwei sehr unterschiedliche Gruppen von Schülerinnen und Schülern gab: Jene mit bereits breitem (selbst angeeignetem) Wissen, welche den Großteil der im Informatikunterricht vermittelten Inhalte schon kannte und somit das Fach eher als uninteressant einstufte versus derjenigen Gruppe, für die dieses Fach grundsätzlich neue Inhalte enthielt. Durch die Einführung eines einheitlichen, curricularen Informatikunterrichts ab der 6. Jahrgangsstufe im G8 scheint der Schulunterricht nun vor allem für die grundsätzlich Informatik-affinen Schülerinnen und Schüler (vgl. [GB12]) mehr interessante Inhalte zu bieten, so dass dem Informatikunterricht tatsächlich eine homogenisierende und aufklärerische Funktion (wie von Magenheim gefordert (vgl. [MS05]) bestätigt werden kann. Einschränkend müssen die Ergebnisse allerdings so interpretiert werden, dass die Wahrnehmung des Unterrichts (sowohl Interessantheitsgrad, wie auch Neuigkeitsgehalt), je nach (vermutlich) bereits vorhandener Affinität zu einem späteren Studien- bzw. Berufswunsch divergiert (siehe Ergebnisse in der Vergleichsgruppe der Erstsemester EEI). Dies zeigt aber auch, dass die Einführung des Schulfachs Informatik gerade für die Zielgruppe der Informatik-affinen Schülerinnen und Schüler positive Effekte hat: Aus den weiteren Ergebnissen (Fragen 4 und 5) kann angenommen werden, dass nun erstmals auch Schülerinnen und Schüler zu einem Informatikstudium tendieren, welche sich das Studium bislang (z. B. aus fachlicher Unwissenheit) nicht zugetraut haben (vgl. [En08]). Gerade aus den Ergebnissen der Frage 5 wird deutlich, dass die bisherigen G9-Absolventen, welche sich für ein Informatikstudium entschieden haben, nur durch außerschulisch geprägte Erfahrungen zu ihrer Studienwahl gelangt sind. Falls ein Informatikunterricht stattfand, waren dessen Inhalte für einen Großteil der Schülerinnen und Schüler weder ausreichend noch befriedigend.

Erstmalig gaben G8-Absolventen, welche sich für ein Informatikstudium entschieden haben, Einflussfaktoren aus dem schulischen Bereich an. Der Einfluss außerschulischer Faktoren ist aber weiterhin sehr hoch. Die Studie zeigt somit auch, dass sich gerade Informatik-affine Schülerinnen und Schüler nach wie vor viel IT-Wissen außerhalb der Schule aneignen. Allerdings bescheinigt die Studie deutlich (vgl. Frage 6), dass der Informatikunterricht für die Zielgruppe der Informatik-affinen Schülerinnen und Schüler eine wichtige Entscheidungsgrundlage für die Wahl des Informatikstudiums darstellt.

4 Fazit und Ausblick

Der Ergebnisse der durchgeführten Studie zeigen vor allem ein divergentes Bild der Wahrnehmung des Informatikunterrichts bei Erstsemestern unterschiedlicher Studienrichtungen, was vermuten lässt, dass sich bei Schülerinnen und Schüler bereits frühzeitig Affinitäten zu bestimmten Studienrichtungen und Berufsfeldern bilden. Sie zeigt jedoch auch, dass ein verpflichtender Informatikunterricht, gerade für die Zielgruppe der Infor-

matik-affinen Schülerinnen und Schüler, eine positiv verstärkende Wirkung für die spätere Studienwahl hat. Ein Befund, der empirisch noch weiter zu festigen wäre. Da die Studien- und Berufswahl ein längerfristiger, individueller und multi-motivational begründeter Prozess ist, müssen weitere endogene und exogene Faktoren (wie z. B. Identifikation mit dem Berufsbild, Arbeitsmarktsituation, finanzielle Interessen, Einflüsse vom Elternhaus, vgl. z. B. [Sc08]) berücksichtigt werden. Die Einführung eines verpflichtenden Informatikunterrichts kann nach dieser Studie als eine notwendige aber lange noch nicht hinreichende Maßnahme bezeichnet werden, um mehr Schülerinnen und Schüler für ein Studium im MINT-Bereich zu gewinnen.

Die Studie zeigt, dass bislang viel Potential im schulischen Bereich „schlummerte“ und ungenutzt blieb. Die Einführung eines verpflichtenden Schulfaches Informatik kann daher grundsätzlich als zielführend betrachtet werden, es sind aber weitere Bemühungen im schulischen Umfeld nötig. So könnten Universitäten und Hochschulen mehr unterstützende Angebote für Schulklassen und Lehrer im MINT-Bereich bereitstellen (z. B. Schülerlabore, betreute Projekt- und P-Seminare, Internet-Lernportale, Schul-Wettbewerbe, Ferien-Workshops, Science-Center, Campustage für Schüler und Eltern, Lehrerfortbildungen, etc.). In den USA gibt es in diesem Zusammenhang an vielen Universitäten sogenannte „Outreach-Units“, mit dem Ziel, eine intensivere Zusammenarbeit zwischen Hochschulen und Gymnasien zu ermöglichen. Hierzu sind weitere Untersuchungen angedacht, um herauszufinden, welche weiteren Angebote seitens der Universitäten zielführend sind, dem drohenden MINT-Fachkräftemangel entgegenzuwirken.

Literaturverzeichnis

- [AKP12] Anger, C.; Koppel, O.; Plünnecke, A.: MINT - Herbstreport 2012. Berufliche MINT-Qualifikationen stärken. Köln, 2012, URL: http://www.mintzukunftschaften.de/uploads/media/FINAL_MINT-Herbstreport_2012.pdf, gepr. 17.05.2013.
- [BD08] Beaubouef, T.; McDowell, P.: Computer Science: Student myths and misconceptions. In: Journal of Computing Sciences in Colleges 23 (6), 2008, S. 43–48.
- [Ca06] Carter, L.: Why students with an apparent aptitude for computer science don't choose to major in computer science. In: ACM (ed.): Proceedings of the thirty-seventh SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2006). ACM Press, New York, 2006, S. 27.
- [En08] Engeser, S. et al.: Abschlussbericht zur Untersuchung Studienwahl Informatik. Technische Universität, München. Lehrstuhl für Psychologie, Juli, 2008. URL: http://www.psy.wi.tum.de/Docs/Studienwahl_Informatik-Abschlussbericht.pdf, gepr. 17.05.2013.
- [GB12] Götz, C.; Brinda, T.: Sind soziale Netzwerke geeignet, um darin für Informatikstudiengänge zu werben? In: GI (Hrsg.): Praxisband zur GI-Fachtagung "Informatik für eine nachhaltige Zukunft - HDI 2012", 2012, S. 137–148.
- [Hu05] Hubwieser, P.: Von der Funktion zum Objekt - Informatik für die Sekundarstufe 1 (Hauptvortrag INFOS 2005). In Friedrich S. (Hrsg.): Unterrichtskonzepte für die informatische Bildung. GI-Edition Lecture Notes in Informatics. P-60, S. 27–41, 2005.
- [Hu07] Hubwieser, P.: Informatische Bildung in Bayern. LOG IN Nr. 145, Seiten 6–8, 2007.
- [Hu12] Hubwieser, P.: Computer Science Education in Secondary Schools - The Introduction of a New Compulsory Subject. In: (ACM) Transaction on Computer Education 12 (4), 2012, S. 1–41.

- [Kn11] Knobelsdorf, M.: Biographische Lern- und Bildungsprozesse im Handlungskontext der Computernutzung, Dissertation, FU Berlin, Fachbereich Mathematik und Informatik, 2011.
- [KP09] Koppel, O.; Plünnecke, A.: Fachkräftemangel in Deutschland. Bildungsökonomische Analyse, politische Handlungsempfehlungen, Wachstums- und Fiskaleffekte (IW-Analysen, 46). Dt. Inst.-Verl, Köln, 2009.
- [Le12] Leszczensky, M. et al.: Bildung und Qualifikation als Grundlage der technologischen Leistungsfähigkeit Deutschlands. Bericht des Konsortiums "Bildungsindikatoren und technologische Leistungsfähigkeit" (Studien zum deutschen Innovationssystem, 01-2012). HIS, Hannover, 2012.
- [MS05] Magenheimer, J.; Schulte, C.: Erwartungen und Wahlverhalten von Schülerinnen und Schülern gegenüber dem Schulfach Informatik – Ergebnisse einer Umfrage. In: Friedrich, S. (Hrsg.): Unterrichtskonzepte für informatische Bildung. Köllen, Bonn, 2005, S. 111-121.
- [MW06] Maaß, S.; Wiesner, H.: Programmieren, Mathe und ein bisschen Hardware ... Wen lockt dies Bild der Informatik? In: Informatik-Spektrum 29 (2), 2006, S. 125–132.
- [Pr11] Prognos AG (Hrsg.): Arbeitslandschaft 2030: Die Studie zum Arbeitskräftemangel in Deutschland und Bayern, September 2011. URL: http://www.prognos.com/fileadmin/pdf/publikationsdatenbank/110930_Neuaufgabe_Arbeitslandschaft_2030.pdf, gepr. 17.05.2013
- [Sc08] Schwanzer, A. D.: Determinanten der Berufswahl. Die Berufswahl von Abiturientinnen und Abiturienten und Konsequenzen einer Berufsentscheidung bei konfligierenden Determinanten. Dissertation Fachbereich Pädagogik, Univ. Erlangen-Nürnberg 2008. Kovač, Hamburg, 2008.

InTech – Eine wirksame Langzeitfortbildung für Informatik-Lehrkräfte

Claudia Hildebrandt und Ira Diethelm
Carl von Ossietzky Universität

Fakultät II - Didaktik der Informatik
26111 Oldenburg
claudia.hildebrandt@uni-oldenburg.de
ira.diethelm@uni-oldenburg.de

Abstract: Angesichts der MINT-Bedeutung für die Gesellschaft im Privaten wie im Berufsleben ist es wichtig, einen informatisch-technischen Bereich für möglichst alle Schülerinnen und Schüler bereitzustellen, in dem sie die Möglichkeit haben, technikbezogene Inhalte und Arbeitsweisen auszuprobieren, ggf. darin Talente zu entdecken und sich vor allem informiert frei entscheiden zu können. Damit Informatiklehrerinnen und -lehrer nachhaltige Grundlagen für einen verantwortungsvollen Umgang mit Informationstechnologie vermitteln können, benötigen sie neben der fachlichen Kompetenz ein Curriculum und geeignetes Unterrichtsmaterial. Da in Niedersachsen Informatik in der Sekundarstufe I noch nicht etabliert ist und es kaum ausgebildete Lehrkräfte für das Fach gibt, wurde dies in einer Lern- und Arbeitsgemeinschaft (einem sog. Lehrerset) angegangen und wissenschaftlich begleitet. Dieser Artikel geht den folgenden Fragen nach: Inwiefern verändert sich der Informatikunterricht im Laufe der drei Projektjahre aus der Sicht der Lehrkräfte? Gibt es einen Zusammenhang zwischen diesem projektbezogenen Einfluss auf den Informatikunterricht und der wahrgenommenen Qualität des Lehrersets?

1 Einleitung

Im niedersächsischen Projekt InTech – Informatik mit technischen Aspekten – hatten sich zur Etablierung des Faches Informatik in der Sek. I Lerngemeinschaften, bestehend aus Informatiklehrkräften verschiedenster Schulformen und Wissenschaftlern gebildet, die unter anderem Fortbildungen gestalteten, Unterrichtseinheiten entwickelten und erprobten. Begleitend wurde untersucht, ob diese Langzeitfortbildung und u.a. ihre wahrgenommene Qualität positive Auswirkungen auf den Informatikunterricht hat.

In Kapitel 2 werden kurz die Merkmale wirksamer Lehrerfortbildungen umrissen. Es folgt die allgemeine Beschreibung des Projekts und seines Konzepts (siehe Kapitel 3). Die Forschungsfragen werden in Kapitel 4 dargestellt und die Planung und der Ablauf der quantitativen Datenerhebung in Kapitel 5 erläutert. Anschließend werden die Resultate zu den Forschungsfragen diskutiert (siehe Kapitel 6) und die Ergebnisse in Kapitel 7 zusammengefasst.

2 Stand der Forschung

Nicht erst seit den jüngsten Ergebnissen John Hatties ([Hat12]) steht die Lehrkraft im Fokus der Verbesserung von Unterricht. Schon in [Grä08] wurde der Frage nachgegangen, wie Lehrerfortbildungen zu gestalten sind, die weiterreichende Veränderungen des Unterrichts bewirken. Es geht nicht nur darum, bestimmte Unterrichtsmethoden zu nutzen, sondern ebenfalls um größere Problem- und Verständnisorientierung oder einen anderen Umgang mit Schülervorstellungen bzw. Fehlern (vgl. [Grä08], S. 64f).

Nach Gräsel, Pröbstel, Freienberg und Parchmann ([GFPF06], S. 311) haben folgende Merkmale Auswirkungen auf eine Veränderung des Unterrichts:

Langfristigkeit: Die Dauer der Fortbildung sollte mindestens drei bis sechs Monate betragen und Theoriephasen sollten sich mit Praxisphasen abwechseln. Handlungsrountinen wie Überzeugungen (beliefs) können nur durch längerfristige Interventionen verändert werden. „Als besonders fruchtbar für die Veränderung unterrichtsbezogener Kognitionen erweist es sich, wenn Lehrkräfte Diskrepanzen zwischen eigenen Überzeugungen und eigenem Handeln erleben und wenn sie mit Argumentationsmustern, subjektiven Theorien und Überzeugungen anderer Lehrkräfte konfrontiert werden. Damit steigt auch die Wahrscheinlichkeit, dass Lehrerfortbildungen Effekte auf Schülerleistungen haben“ ([Lip04], S. 474).

Situierung: Die Inhalte der Fortbildungen müssen einen Bezug zu den Bedingungen an den Schulen (z.B. Lehrpläne) und die Bedürfnisse der Lehrkräfte aufweisen und offen für Veränderungen und Adaptionen sein.

Anregung zur Kooperation unter Kolleginnen und Kollegen: Die Zusammenarbeit forciert eine diskursive Auseinandersetzung und fördert damit eine nachhaltige Änderung der Unterrichtspraxis.

Von den Lehrkräften selbst wahrgenommene Veränderungen der Unterrichtspraxis scheinen positiv mit den Leistungen der Schülerinnen und Schüler zusammen zuhängen (vgl. [CH00], S. 324f).

3 Der Modellversuch InTech

3.1 Phase I

Die erste Phase des Modellversuchs wurde von 2005 bis 2008 durchgeführt, initiiert von Vera Reineke, Niedersächsisches Kultusministerium, und Prof. Dr. Eckart Modrow, Didaktik der Informatik der Universität Göttingen, der darüber hinaus die Koordination des Projektes übernahm. Die Stiftung NiedersachsenMetall finanzierte die technische Ausstattung der Schulen für den Modellversuch. An sechs Gymnasien wurde erprobt, wie technisch orientierter Informatikunterricht in den Schuljahrgängen 7 bis 9 erteilt werden kann (vgl. [MR08] und [Mod11]).

3.2 Phase II

Um die positive Erfahrungen aus der ersten Phase des Modellversuchs weiterzutragen und zu verfestigen sowie eine Begleitforschung durchzuführen, schloss sich von 2009 bis 2012 Phase II an. Sie bestand aus zwei Teilprojekten, die von der Universität Göttingen und zusätzlich von der Universität Oldenburg umgesetzt wurden. Am 01.08.2009 startete dieses Projekt mit 13 Schulen in der Weser-Ems-Region und 10 Schulen aus dem gesamten östlichen Niedersachsen. Das Land förderte diesen Modellversuch erneut. Die Sachausstattung der teilnehmenden Schulen wurde zu einem großen Teil von dem Arbeitgeberverband NORDMETALL, der Stiftung des Verbands der Metall- und Elektroindustrie - VME und der Stiftung NiedersachsenMetall übernommen.

Voraussetzung für die teilnehmenden Schulen war, Informatik in der Mittelstufe als Unterrichtsfach anzubieten oder einrichten zu wollen und mindestens zwei qualifizierte Informatiklehrkräfte an der Schule zu haben. Die Vertreter der jeweiligen Schulen sollten an den regelmäßig alle 6 - 8 Wochen stattfindenden Projekttreffen teilnehmen. Um die Nachhaltigkeit dieses Modellversuchs zu gewährleisten, wurden folgende Ziele bezüglich der Lehrkräfte verfolgt (vgl. [HD12]): Austausch von Erfahrungen, Etablierung eines Netzwerkes der Informatiklehrerinnen und -lehrer unterschiedlicher Schulen und Schulformen, Entwicklung, Test, Reflexion und Weiterentwicklung von schüler- und kontextorientiertem Unterrichtsmaterial, Veröffentlichung des Unterrichtsmaterials über das Internet und weitere Lehrerfortbildungsmaßnahmen.

Die folgenden Ausführungen beziehen sich immer auf das Teilprojekt Oldenburg.

3.3 Das Konzept

Die beschriebenen Ziele wurden mithilfe der sog. symbiotischen Implementationsstrategie realisiert (vgl. [FGPP04], S. 242-244). Die hier im Folgenden näher betrachtete Lerngemeinschaft des Teilprojekts Oldenburg umfasste ursprünglich 24 Lehrkräfte von 13 verschiedenen Schulen und drei Wissenschaftlerinnen (vgl. [DHK09]).

Während der drei Projektjahre wechselten sich immer wieder Phasen der Instruktion, der Konstruktion und Reflexion ab. Es gab Inputs, vor allem zu fachlichen Themen und zu Methoden und Medien, sowohl von externen Experten als auch von gruppeninternen Experten. Die Fortbildungsmaßnahmen handelten unter anderem von der Einführung in die Roboterkonstruktion und Robotersteuerung mit Lego Mindstorms, Programmiersprachen für Lego-Mindstorms NXT, Datenschutz und Urheberrecht im Informatikunterricht, über die Funktionsweise des Internets, von Einstiegen in die Algorithmik in der Sek 1 am Beispiel der Automatisierung realer Miniwelten, von Netzwerken als Thema im Jahrgang 10, von Technikunterricht und Informatik, vom Programmieren mit Scratch, Karol und Etoys und über IT-Support an Schulen.

Die Lehrkräfte nutzten die Bildungsstandards Informatik für die Sekundarstufe I der GI vor allem in der Konstruktionsphase. Darin entwickelten die Lehrkräfte selbstständig Unterrichtsmodule, notierten sie entsprechend einer einheitlichen Vorlage, die die Struk-

turierung von Unterrichtseinheiten erleichtern und vereinheitlichen sollte, reflektierten, evaluierten und optimierten sie gegenseitig. Die entstandenen Unterrichtsmaterialien, die z.T. projektorientiert, schüler- und kontextorientiert sind, wurden über eine gemeinsame Plattform (IServ) ausgetauscht. Zu den Themen Robotik, Netzwerke, Kryptographie, Programmierung, digitale Elektronik, Audiotbearbeitung, Bildbearbeitung, Urheberrecht und Datenschutz und Wettbewerbe befinden sich Unterrichtsmaterialien auf www.intechportal.de. Insgesamt wurde immer wieder für die Lehrkräfte Raum gegeben eigene Anliegen zu thematisieren und Erfahrungen auszutauschen.

In den Reflektionsphasen erfolgte eine Bewertung der Sitzungen u.a. in Form eines Fragebogens sowie die Planung und Organisation zukünftiger Arbeitstreffen. Höhepunkt eines jeden Projektjahres war der von den Lehrkräften selbst organisierte und von den Wissenschaftlern unterstützte InTech-Cup, ein Roboter-Wettbewerb, bei dem die Schülerinnen und Schüler die Möglichkeit haben, ihre Fähigkeiten zum kooperativen, selbstständigen und zielgerichteten Arbeiten auch außerhalb der Schule unter Beweis zu stellen.

Der Erfolg der Implementation dieses Projekts und seines Konzeptes war von zahlreichen Faktoren abhängig wie den Einstellungen, Überzeugungen und Kompetenzen der Lehrkräfte, der Kooperations- und Kommunikationskultur und der Qualität der Lerngemeinschaften (vgl. [GP04], S. 208-209). Im Folgenden werden die Aspekte "Qualität der Setarbeit und Unterstützung für den Informatikunterricht aus der Sicht der Lehrkräfte" näher betrachtet. Ergebnisse der Evaluation der Schülerfragebögen sind in [HD12] zu finden.

4 Forschungsfragen

Ein zentrales Anliegen des Projektes ist, die professionelle Kompetenz der Lehrkräfte zu steigern, um damit positiv auf die Motivation, das Interesse und die Leistung der Schülerinnen und Schüler einzuwirken.

1. Inwiefern verändert sich der Informatikunterricht im Laufe der drei Projektjahre aus der Sicht der Lehrerinnen und Lehrer?
Die langfristige und situierte Kooperation der teilnehmenden Lehrkräfte unterstützt und begleitet durch Wissenschaftler sollte analog zu den in beispielsweise [Lip04] und [Grä08] angeführten Ergebnissen eine für die Lehrer wahrnehmbare positive Auswirkung auf ihren Unterricht haben. Dabei werden die Unterrichtsmerkmale Methodenvielfalt, Schüler- und Kontextorientierung untersucht.
2. Inwiefern hängen die von den Lehrkräften wahrgenommene Qualität des Lehrersets und die Unterstützung für den Informatikunterricht zusammen?
Wir vermuten, dass neben den von [GPF06] (S. 311) angeführten Merkmalen ebenfalls die Qualität der Arbeitsgemeinschaften und die Veränderungen des Unterrichts in einem Zusammenhang stehen. Je höher die Qualität der Setarbeit eingestuft wird, desto größer ist der Einfluss auf den Informatikunterricht.

5 Forschungsmethode

5.1 Planung und Ablauf der Erhebung

An unserem Teilprojekt nahmen 24 Lehrkräfte (17% w, 83% m) und ca. 300 Schülerinnen und Schüler teil. Zur Erhebung der Daten im Rahmen der empirischen Forschung wurden Onlinefragebögen verwendet. Die jeweilige Anzahl der Teilnehmer an der Lehrer-Umfrage schwankt zwischen 19 und 24 über die Erhebungszeitpunkte.

Davon haben 13 - anhand von individuellen Codes feststellbar - an allen vier Lehrerumfragen teilgenommen. Um die Entwicklung über die Zeit zu beobachten, werden diese 13 Datensätze der Lehrkräfte (2 w, 11 m) untersucht. Zum Zeitpunkt t_4 gaben drei Lehrkräfte an zwischen 30 und 39 Jahre alt zu sein, sechs zwischen 40 und 49 Jahre, drei zwischen 50 und 59 und eine Lehrkraft zwischen 60 und 69 Jahre.

Bei der formalen Kompetenz, um Informatikunterricht geben zu können, nannten fünf Lehrkräfte Informatik als Erweiterungsfach, also als drittes Fach, zu haben, vier nannten Weiterbildungsmaßnahmen (z.B. vlin), lediglich eine Lehrkraft hat Informatik als Erst- oder Zweitfach studiert und das Referendariat im Fach Informatik absolviert. Selbst viel beigebracht haben sich nach eigenen Angaben sieben Lehrkräfte. Mehrfachantworten waren bei dieser Frage möglich. Zum Zeitpunkt t_4 besaßen vier Personen Unterrichtserfahrung im Fach Informatik von 3-5 Jahre, fünf von 6-10 Jahren, zwei von 11-15 Jahren und zwei von über 25 Jahren. Bei der Anzahl an Informatikkolleginnen und -kollegen an der Schule gab es folgende Resultate: Eine Kollegin/ein Kollege wurde 2-mal angegeben, zwei Kolleginnen/Kollegen gab es 3-mal, drei 2-mal, vier 2-mal, fünf 1-mal (3-mal gab es zum Zeitpunkt t_4 keine Angaben).

5.2 Definition einer Auswahl der lehrerbezogenen Konstrukte

Eine zentrale Forschungsfrage des Projekts ist, ob sich Veränderungen zeigen, die auf den Implementationserfolg hinweisen. Dabei wird untersucht, ob schülerorientierter Unterricht stattfindet, größere Methodenvielfalt und mehr Kontextorientierung zu verzeichnen sind (siehe Kapitel 4). Um Informationen zu diesen Themengebieten zu bekommen, wurden unter anderem folgende Konstrukte aufgestellt.

Methodenvielfalt: Dieses Konstrukt stellt dar, ob im Unterricht unterschiedliche (der Situation angepasste) Lernmethoden Anwendung finden.

Schülerorientierung: Dieses Konstrukt bedeutet, unterrichtliches Lehren und Lernen, also die Zielauswahl, Inhaltsbestimmung und Methodengestaltung des Unterrichts, an den Schülerinnen und Schülern auszurichten (vgl. [Sch01]).

Kontextorientierung: Dieses Konstrukt untersucht, ob für den Lernenden nachvollziehbare und interessante Zusammenhänge aus seiner Lebenswirklichkeit Ausgangspunkt des unterrichtlichen Lehrens und Lernens sind.

Unterstützung für den Informatikunterricht: Das Konstrukt beinhaltet die Möglichkeiten, die das Projekt bieten, um u. a. Hilfen, Anregungen, Unterrichtsmaterialien für den eigenen Informatikunterricht zu bekommen bzw. gemeinsam zu entwickeln.

Qualität der Setarbeit: Dieses Konstrukt umfasst die allgemeine Beurteilung des Sets, von der sozialen Atmosphäre, über die Kooperation der Kolleginnen und Kollegen untereinander bis hin zur Arbeit der Koordination und zu dem Engagement der Betreuerinnen.

Eine Übersicht über die erfassten Konstrukte mit Beispielitems und den Reliabilitäten der Skalen (Cronbachs Alpha) sind in Tabelle 1 zu finden.

Tabelle 1: Konstrukte Methodenvielfalt, Schüler- und Kontextorientierung

Konstrukt	Items	Beispielitem	Cronbachs Alpha
Methodenvielfalt	3	In meinem Informatikunterricht verwende ich ein breites Spektrum an Methoden.	t ₀ : 0,794 t ₁ : 0,767 t ₂ : 0,797 t ₄ : 0,736
Schülerorientierung	6	In meinem Informatikunterricht gibt es viele Phasen, in denen die Schüler/-innen einen Freiraum bekommen, mit dem sie eigenverantwortlich umgehen müssen.	t ₀ : 0,716 t ₁ : 0,804 t ₂ : 0,696 t ₄ : 0,716
Kontextorientierung	5	Ich berücksichtige meistens lebensnahe und schülerorientierte Themen bei der Planung von Informatikunterrichtseinheiten.	t ₀ : 0,746 t ₁ : 0,834 t ₂ : 0,702 t ₄ : 0,924
Unterstützung für Unterricht	7	Die Arbeit im Projekt ermöglicht mir, gemeinsam Unterrichtsmaterial für den Informatikunterricht zu entwickeln.	t ₀ : 0,772 t ₁ : 0,142 t ₂ : 0,786 t ₄ : 0,802
Qualität der Setarbeit	4	Ich arbeite mit den Kolleginnen und Kollegen im Projekt produktiv zusammen.	t ₀ : 0,732 t ₁ : 0,819 t ₂ : 0,903 t ₄ : 0,824

5.3 Instrumente

Der genutzte Lehrerfragebogen basiert auf den im Rahmen des ChiK-Projekts¹ entwickelten Fragebögen. Die Konstrukte und Items wurden entsprechend der Anforderungen des Projekts und der speziellen Bedingungen der Informatiklehrkräfte angepasst. Die Untersuchung der Konstrukte erfolgt anhand von Items. Die Lehrkräfte werden jeweils nach dem

¹Bei den Fragebögen des ChiK-Konzepts wurde in Teilen auf andere Projekte wie z. B. das SINUS-Projekt zurückgegriffen, vgl. [Ost04] und [GFSZ]

Grad ihrer Zustimmung bzw. Ablehnung zu den Items befragt. Die Antwortskalen der Items sind sechsstufig, von "1 = trifft überhaupt nicht zu" bis "6 = trifft völlig zu" (Likert Skala). Tabelle 1 zeigt die Konstrukte, Beispielitems und die Cronbachs Alpha. Cronbachs Alpha ist eine Maßzahl, die feststellt, inwieweit eine Gruppe von Test-Items als Messung einer einzelnen Variablen (hier der einzelnen Konstrukte) angesehen werden kann. Das heißt, sie gibt an, inwiefern verschiedene Items im Grunde das gleiche messen. Der Wert für Cronbachs Alpha sollte 0,7 oder mehr erreichen. In Anbetracht der Tatsache, dass an der Lehrerumfrage zum Zeitpunkt t_2 lediglich 19 Personen teilnahmen, ist ein Wert für Cronbachs Alpha von 0,696 jedoch akzeptabel. 0,142 hingegen ist ein zu schlechter Wert, um bei den untersuchten Items von einem Konstrukt sprechen zu können. Zu den anderen Umfragezeitpunkten sind die Werte für Cronbachs Alpha allerdings akzeptabel, so dass alle dargestellten Konstrukte untersucht werden.

Antworten auf einer Likert-Skala sind typischerweise ordinal- beziehungsweise rangskaliert. Da die Antwortmöglichkeiten allerdings durch eine äquidistante Skala visualisiert werden, sollten die Testteilnehmer die verschiedenen Antwortmöglichkeiten als äquidistant wahrnehmen, so dass wir das Intervallskalenniveau verwenden. Hier liegt die Erkenntnis zugrunde, dass die Ergebnisse der statistischen Tests nur unwesentlich durch die Interpretation von eher ordinalskalierten Daten als intervallskaliert verfälscht werden ([HSE10], S. 67).

6 Ergebnisse

6.1 Forschungsfrage 1

Eine Forschungsfrage des Projekts ist, ob sich Veränderungen bezüglich des Informatikunterrichts aus der Wahrnehmung der Lehrkräfte zeigen, die auf den Implementationserfolg hinweisen. Dabei wird untersucht, ob mehr schüler- und kontextorientierter Unterricht zu verzeichnen sind und die Methodenvielfalt zunimmt. Zur Untersuchung der ersten Forschungsfrage *Inwiefern verändert sich der Informatikunterricht im Laufe der drei Projektjahre aus der Sicht der Lehrer?* werden die Konstrukte Methodenvielfalt, Schüler- und Kontextorientierung zu den Erhebungszeitpunkten t_0 , t_1 , t_2 und t_4 untersucht. Die Skalenmittelwerte und Standardabweichungen bezüglich dieser Konstrukte sind den Tabellen 2 und 3 zu entnehmen. Der Tabelle 2 ist zu entnehmen, dass die Skalenmittelwerte der drei

Tabelle 2: Skalenmittelwerte m der Unterrichts-Konstrukte

Konstrukt	m_0	m_1	m_2	m_4	$m_4 - m_0$
Methodenvielfalt	4,08	3,97	4,15	4,31	+ 0,23
Schülerorientierung	3,49	3,38	3,35	3,65	+ 0,16
Kontextorientierung	3,43	3,47	3,70	3,75	+ 0,32

Konstrukte leicht angestiegen sind. Somit könnte man schließen, dass die Lehrkräfte der Meinung sind, Methodenvielfalt sowie Schüler- und Kontextorientierung ein wenig mehr

Tabelle 3: Standardabweichungen σ für die Skalenmittelwerte der Unterrichts-Konstrukte

Konstrukt	σ_0	σ_1	σ_2	σ_4
Methodenvielfalt	0,59	0,67	0,74	0,78
Schülerorientierung	0,69	0,72	0,54	0,91
Kontextorientierung	0,76	0,81	0,64	1,10

im Unterricht einzusetzen als zu Beginn des Projektes. Dies kann auf die konstruktive Arbeit im Projekt zurückzuführen sein, da die Lehrerinnen und Lehrer vom Unterrichtsmaterial und den Erfahrungen der Kolleginnen und Kollegen profitieren und deutet auf einen Implementationserfolg hin. Diese Selbstwahrnehmung der Lehrkräfte kann sich positiv auf die Schülerleistungen auswirken (vgl. [Lip04], S. 472). Signifikant sind diese Ergebnisse nach einer statistischen Überprüfung nicht, was auch an dem geringen Stichprobenumfang von 13 Probanden liegen kann.

6.2 Forschungsfrage 2

Zur Untersuchung der zweiten Forschungsfrage *Inwiefern hängen die von den Lehrkräften wahrgenommene Qualität der Arbeitsgemeinschaften und die Unterstützung für den Informatikunterricht zusammen?* werden die Konstrukte Unterstützung für den Informatikunterricht und Qualität der Setarbeit herangezogen (siehe Tabelle 4 und 6). Der Tabelle 4 ist zu entnehmen, dass sich der Skalenmittelwert für das Konstrukt Unterstützung für Informatikunterricht über die drei Jahre hinweg kaum verändert und mit einem Wert um 4,5 auf einer Skala von 1 bis 6 relativ hoch ist. Die Lehrkräfte sind fast durchweg der Meinung, dass die Setarbeit ihren Informatikunterricht positiv beeinflusst und sie Unterstützung bekommen. Sie können sich austauschen und von den Erfahrungen und Unterrichtsmaterialien profitieren. Dies ist gerade für Informatiklehrkräfte von großer Bedeutung, da nur ein teilnehmender Lehrer Informatik als Erst- oder Zweifach studiert hat und das Referendariat im Fach Informatik absolvierte. Darüber hinaus gibt es nur sehr wenige Fachkolleginnen und -kollegen an den Schulen, mit denen man sich austauschen könnte. Somit sind schulübergreifende Netzwerke zur Erhöhung der Lehrerkompetenz unabdingbar.

Tabelle 4: Skalenmittelwerte m der Set-Konstrukte

Konstrukt	m_0	m_1	m_2	m_4
Unterstützung für Informatikunterricht	4,53	4,16	4,69	4,48
Qualität der Setarbeit	4,94	5,10	5,23	5,23

Tabelle 5: Standardabweichungen σ für die Skalenmittelwerte der Set-Konstrukte

Konstrukt	σ_0	σ_1	σ_2	σ_4
Unterstützung für Informatikunterricht	0,77	0,38	0,63	0,62
Qualität der Setarbeit	0,72	0,63	0,50	0,58

Die Lehrkräfte schätzen die Qualität der Setarbeit insgesamt mit durchschnittlich über 5,1 als sehr hoch ein. Der Skalenmittelwert nimmt von Beginn des Projekts bis zum Ende des Projekts leicht zu. Signifikant ist dieser Anstieg nicht.

Tabelle 6 zeigt, dass es ab Erhebungszeitpunkt t_1 , also ab Ende des ersten Projektjahres einen deutlichen Zusammenhang zwischen der wahrgenommenen Qualität der Setarbeit und der Unterstützung für den Informatikunterricht gibt. Am Anfang des Projektes kann es diesen Zusammenhang noch nicht geben, da die teilnehmenden Lehrkräfte gar nicht so genau wissen, was genau auf sie zukommt, da Langzeitfortbildungen eher unbekannt sind. Welche Richtung dieser Zusammenhang hat, kann man aus dem Spearmans Roh Korrelationskoeffizienten nicht ablesen. Um das herauszufinden, wären weitere Untersuchungen notwendig. Auch der Grad der Wirksamkeit einzelner Variablen auf die Veränderung der kognitiven Strukturen lässt sich so nicht angeben.

Tabelle 6: Korrelationen (Spearmans Roh, * $p < 0,01$, ** $p < 0,05$, andere Korrelationen sind nicht signifikant)

	Unterstützung für Informatikunterricht			
Qualität der Setarbeit	t_0 : 0,158	t_1 : 0,583**	t_2 : 0,833*	t_4 : 0,641**

7 Zusammenfassung

Zusammenfassend lässt sich festhalten, dass der Modellversuch die Merkmale wirksamer Lehrerfortbildungen (siehe Kapitel 2) berücksichtigt. Darüber hinaus wurde viel Wert darauf gelegt, dass ein Austausch von Erfahrungen, die Etablierung eines Netzwerkes der Informatiklehrerinnen und -lehrer über Schulgrenzen und Schulformen hinaus, die Entwicklung von schüler- und kontextorientiertem Unterrichtsmaterial und ihre Veröffentlichung stattfand. Das Netzwerk der Lehrkräfte hat über die Projektdauer hinaus bestand. Der neu entstandene Roboterwettbewerb wird trotz Projektende fortgeführt. Eine weitere Kommunikation der Lehrkräfte ist dabei notwendig und wird vom Land unterstützt.

Aufgrund der positiven Untersuchungsergebnisse lässt sich vermuten, dass die Informatiklehrkräfte neben einer fachlichen Fortbildung, vor allem durch die Kooperation mit anderen Informatiklehrkräften neue Unterrichts Anregungen und vor allem Selbstvertrauen in die eigene Arbeit bekommen haben. Inwiefern tatsächlich eine weitreichende Veränderung bei den Lehrkräften, z.B. bezüglich Schülervorstellungen, Problem- und Verständnisorientierung stattgefunden hat, war nicht feststellbar.

Die Qualität der Setarbeit scheint einen positiven Einfluss auf Veränderungen des Unterrichts zu haben. Allerdings wurde kein signifikanter Anstieg bezüglich der Unterrichtsmerkmale Methodenvielfalt, Kontext- und Schülerorientierung festgestellt. Der Anstieg der Skalenmittelmittelwerte deutet aber auf eine positiven Entwicklung hin.

Damit sich Fortbildungsmaßnahmen in einer Veränderung des Unterrichts niederschlagen, sollte in Zukunft vermehrt darüber nachgedacht werden, Langzeitfortbildungen anzubieten, auch wenn sie zeit- und kostenintensiver sind und von den Lehrkräften eine höhere Motivation abverlangen.

Literatur

- [CH00] D.K Cohen und H.C. Hill. Instructional policy and classroom performance: The mathematics reform in California. *Teachers College Record*, 102(2):294–343, 2000.
- [DHK09] I. Diethelm, C. Hildebrandt und L. Krekeler. Implementation of Computer Science in Context - a research perspective regarding teacher-training. *9th Koli Calling International Conference on Computing Education Research, Finland*, Seiten 96–99, November 2009.
- [FGPP04] A. Fey, C. Gräsel, T. Puhl und I. Parchmann. Implementation einer kontextorientierten Unterrichtskonzeption für den Chemieunterricht. *Unterrichtswissenschaft*, 32(3):238–256, 2004.
- [GFSZ] C. Gräsel, K. Fussangel und J. Schellenbach-Zell. *Skalenhandbuch Lehrer/-innen*. (unpublished).
- [GP04] C. Gräsel und I. Parchmann. Implementationsforschung - oder: der steinige Weg, Unterricht zu verändern. *Unterrichtswissenschaft*, 32(3):196–214, 2004.
- [GPFP06] C. Gräsel, C. Pröbstel, J. Freienberg und I. Parchmann. Anregungen zur Kooperation von Lehrkräften im Rahmen von Fortbildungen. In M. Prenzel und L. Allolio-Näcke, Hrsg., *Untersuchungen zur Bildungsqualität von Schule, Abschlussbericht des DFG-Schwerpunktprogramms BiQua.*, Seiten 310–332. Waxmann, Münster, 2006.
- [Grä08] C. Gräsel. Anregungen zur Kooperation von Lehrkräften im Rahmen von Fortbildungen: unterschiedliche Wege. *Beiträge zur Lehrerbildung*, 26(1):64–70, 2008.
- [Hat12] J. Hattie. *Visible learning for teachers*. Routledge, London, New York, 2012.
- [HD12] C. Hildebrandt und I. Diethelm. The school experiment InTech - How to influence interest, self-concept of ability in Informatics and vocational orientation. In R. Romeike und M. Knobelsdorf, Hrsg., *The 7th Workshop in Primary and Secondary Computing Education WiPSCE 2012*, Hamburg, 2012.
- [HSE10] W. Hussy, M. Schreier und G. Echterhoff. *Forschungsmethoden in Psychologie und Sozialwissenschaften*. Springer, Berlin, 2010.
- [Lip04] F. Lipowsky. Was macht Fortbildungen für Lehrkräfte erfolgreich? Befunde der Forschung und mögliche Konsequenzen für die Praxis. *Die deutsche Schule*, 96(4):462–479, 2004.
- [Mod11] E. Modrow. Informatik als technisches Fach. *LOG IN*, (163/164):38–42, 2011.
- [MR08] E. Modrow und V. Reineke. *Der Modellversuch Intech08 - Informatikunterricht mit technischen Aspekten*. Informationsbroschüre, Stiftung NiedersachsenMetall, 2008. http://www.stiftung-niedersachsenmetall.de/docs/SNM_intech_web.pdf.
- [Ost04] C. Ostermeier. *Kooperative Qualitätsentwicklung in Schulnetzwerken. Eine empirische Studie am Beispiel des BLK-Modellversuchsprogramms "Steigerung der Effizienz des mathematisch-naturwissenschaftlichen Unterrichts" (SINUS)*. Waxmann, Münster, 2004.
- [Sch01] H. Schröder. *Didaktisches Wörterbuch*. München, 2001.

Verzeichnis der Autorinnen und Autoren

Bergner, Nadine, 107
Buchholz, Malte, 35
Buettner, Katrin, 77

Döbeli Honegger, Beat, 10
Dammik, Gregor, 77
Diethelm, Ira, 157

Engbring, Dieter, 117

Friedrich, Steffen, 77

Götz, Christian, 147
Gallenbacher, Jens, 87

Heun, Dominik, 87
Hildebrandt, Claudia, 157
Hubwieser, Peter, 147

Knapp, Thomas, 77
Knobelsdorf, Maria, 45

Lerch, Christopher, 45

Neugebauer, Jonas, 117

Otten, Lars, 127

Przybylla, Mareen, 137

Rätz, Martin, 77
Rabel, Magnus, 57
Rohland, Holger, 77
Romeike, Ralf, 137

Sauck, Tony, 45
Schauer, Hanno, 97
Schellartz, Tim, 107
Schirmer, Ingrid, 21
Schroeder, Ulrik, 107
Schulmeister, Sönke, 67
Schwarzbach, Maximilian, 45
Seifert, Oliver, 45
Strecker, Kerstin, 24

Vahrenhold, Jan, 23

Weinert, Martin, 45